Red Hat OpenShift avec NetApp

NetApp container solutions

NetApp
January 25, 2026

Sommaire

Red Hat OpenShift avec NetApp

NVA-1160 : Red Hat OpenShift avec NetApp

Alan Cowles et Nikhil M Kulkarni, NetApp

Ce document de référence fournit la validation du déploiement de la solution Red Hat OpenShift, déployée via Installer Provisioned Infrastructure (IPI) dans plusieurs environnements de centres de données différents, comme validé par NetApp. Il détaille également l'intégration du stockage avec les systèmes de stockage NetApp en utilisant l'orchestrateur de stockage Trident pour la gestion du stockage persistant. Enfin, un certain nombre de validations de solutions et de cas d'utilisation réels sont explorés et documentés.

Cas d'utilisation

La solution Red Hat OpenShift avec NetApp est conçue pour offrir une valeur exceptionnelle aux clients avec les cas d'utilisation suivants :

- Red Hat OpenShift est facile à déployer et à gérer, déployé à l'aide d'IPI (Installer Provisioned Infrastructure) sur du bare metal, Red Hat OpenStack Platform, Red Hat Virtualization et VMware vSphere.
- Puissance combinée du conteneur d'entreprise et des charges de travail virtualisées avec Red Hat OpenShift déployées virtuellement sur OSP, RHV ou vSphere, ou sur du bare metal avec OpenShift Virtualization.
- Configuration et cas d'utilisation réels mettant en évidence les fonctionnalités de Red Hat OpenShift lorsqu'il est utilisé avec le stockage NetApp et Trident, l'orchestrateur de stockage open source pour Kubernetes.

Valeur commerciale

Les entreprises adoptent de plus en plus les pratiques DevOps pour créer de nouveaux produits, raccourcir les cycles de publication et ajouter rapidement de nouvelles fonctionnalités. En raison de leur nature agile innée, les conteneurs et les microservices jouent un rôle crucial dans le soutien des pratiques DevOps. Cependant, la pratique de DevOps à l'échelle de la production dans un environnement d'entreprise présente ses propres défis et impose certaines exigences à l'infrastructure sous-jacente, telles que les suivantes :

- Haute disponibilité à toutes les couches de la pile
- Facilité des procédures de déploiement
- Opérations et mises à niveau non perturbatrices
- Infrastructure pilotée par API et programmable pour suivre l'agilité des microservices
- Multilocation avec garanties de performance
- Capacité à exécuter simultanément des charges de travail virtualisées et conteneurisées
- Capacité à faire évoluer l'infrastructure de manière indépendante en fonction des demandes de charge de travail

Red Hat OpenShift avec NetApp reconnaît ces défis et présente une solution qui permet de répondre à chaque préoccupation en mettant en œuvre le déploiement entièrement automatisé de RedHat OpenShift IPI dans

l'environnement de centre de données choisi par le client.

Aperçu de la technologie

La solution Red Hat OpenShift avec NetApp comprend les principaux composants suivants :

Plateforme de conteneurs Red Hat OpenShift

Red Hat OpenShift Container Platform est une plateforme Kubernetes d'entreprise entièrement prise en charge. Red Hat apporte plusieurs améliorations à Kubernetes open source pour fournir une plateforme d'applications avec tous les composants entièrement intégrés pour créer, déployer et gérer des applications conteneurisées.

Pour plus d'informations, visitez le site Web d'OpenShift "ici" .

Systèmes de stockage NetApp

NetApp dispose de plusieurs systèmes de stockage parfaits pour les centres de données d'entreprise et les déploiements de cloud hybride. Le portefeuille NetApp comprend les systèmes de stockage NetApp ONTAP, NetApp Element et NetApp e-Series, qui peuvent tous fournir un stockage persistant pour les applications conteneurisées.

Pour plus d'informations, visitez le site Web de NetApp "ici" .

Intégrations de stockage NetApp

Trident est un orchestrateur de stockage open source et entièrement pris en charge pour les conteneurs et les distributions Kubernetes, y compris Red Hat OpenShift.

Pour plus d'informations, visitez le site Web de Trident "ici" .

Options de configuration avancées

Cette section est dédiée aux personnalisations que les utilisateurs réels devront probablement effectuer lors du déploiement de cette solution en production, comme la création d'un registre d'images privé dédié ou le déploiement d'instances d'équilibreur de charge personnalisées.

Matrice de support actuelle pour les versions validées

Technologie	But	Version du logiciel
NetApp ONTAP	Stockage	9.8, 9.9.1, 9.12.1
NetApp Element	Stockage	12,3
NetApp Trident	Orchestration du stockage	22.01.0, 23.04, 23.07, 23.10, 24.02
Red Hat OpenShift	Orchestration des conteneurs	4.6 EUS, 4.7, 4.8, 4.10, 4.11, 4.12, 4.13, 4.14
VMware vSphere	Virtualisation du centre de données	7.0, 8.0.2

Red Hat Openshift

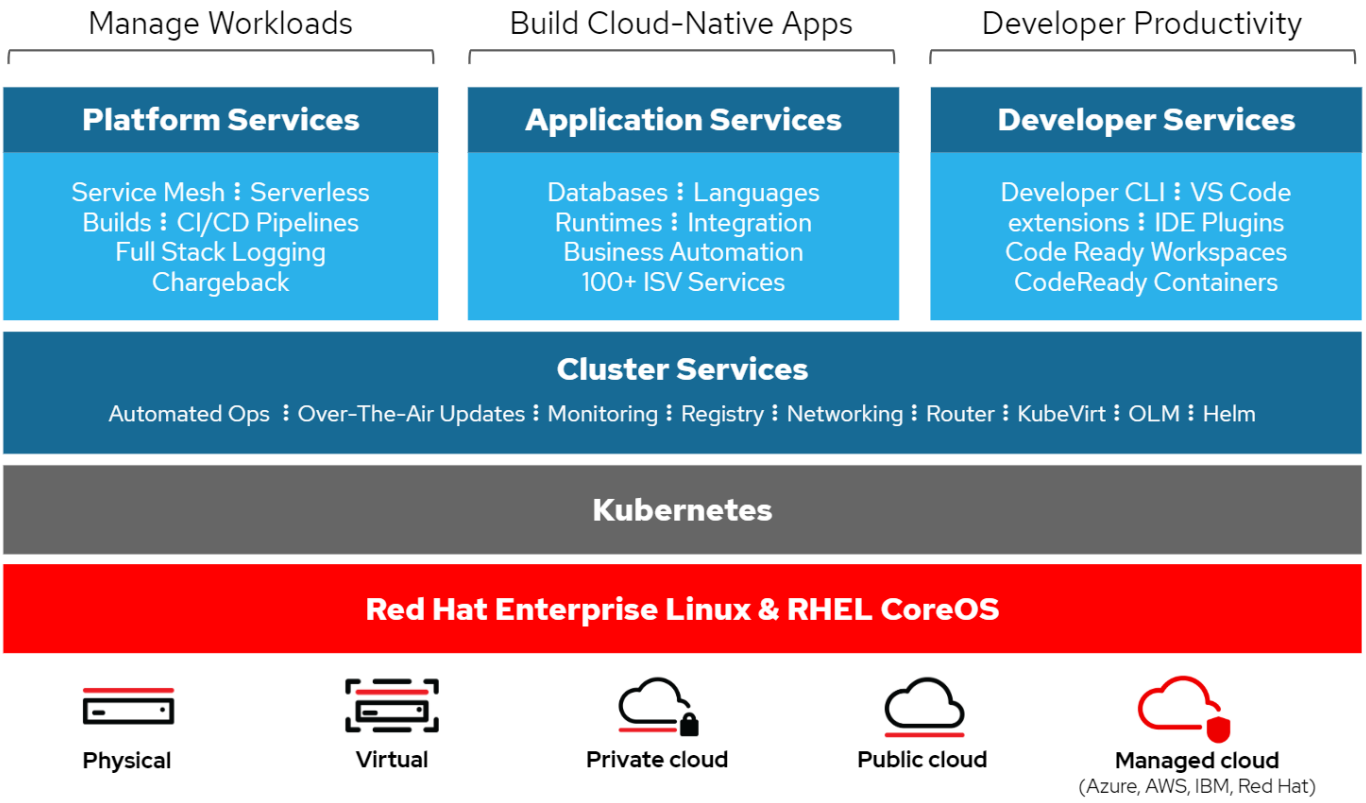
Présentation d'OpenShift

La plateforme de conteneurs Red Hat OpenShift réunit les opérations de développement et informatiques sur une plateforme unique pour créer, déployer et gérer des applications de manière cohérente sur les infrastructures cloud sur site et hybrides. Red Hat OpenShift s'appuie sur l'innovation open source et les normes industrielles, notamment Kubernetes et Red Hat Enterprise Linux CoreOS, la distribution Linux d'entreprise leader au monde conçue pour les charges de travail basées sur des conteneurs. OpenShift fait partie du programme Kubernetes certifié de la Cloud Native Computing Foundation (CNCF), offrant la portabilité et l'interopérabilité des charges de travail des conteneurs.

Red Hat OpenShift offre les fonctionnalités suivantes :

- **Provisionnement en libre-service** Les développeurs peuvent créer rapidement et facilement des applications à la demande à partir des outils qu'ils utilisent le plus, tandis que les opérations conservent un contrôle total sur l'ensemble de l'environnement.
- **Stockage persistant** En fournissant la prise en charge du stockage persistant, OpenShift Container Platform vous permet d'exécuter à la fois des applications avec état et des applications sans état natives du cloud.
- **Intégration continue et développement continu (CI/CD)** Cette plateforme de code source gère les images de build et de déploiement à grande échelle.
- **Normes open source** Ces normes intègrent l'Open Container Initiative (OCI) et Kubernetes pour l'orchestration des conteneurs, en plus d'autres technologies open source. Vous n'êtes pas limité à la technologie ou à la feuille de route commerciale d'un fournisseur spécifique.
- **Pipelines CI/CD** OpenShift fournit une prise en charge prête à l'emploi pour les pipelines CI/CD afin que les équipes de développement puissent automatiser chaque étape du processus de livraison de l'application et s'assurer qu'elle est exécutée à chaque modification apportée au code ou à la configuration de l'application.
- **Contrôle d'accès basé sur les rôles (RBAC)** Cette fonctionnalité fournit un suivi des équipes et des utilisateurs pour aider à organiser un grand groupe de développeurs.
- **Création et déploiement automatisés** OpenShift offre aux développeurs la possibilité de créer leurs applications conteneurisées ou de laisser la plateforme créer les conteneurs à partir du code source de l'application ou même des binaires. La plateforme automatise ensuite le déploiement de ces applications sur l'infrastructure en fonction des caractéristiques définies pour les applications. Par exemple, quelle quantité de ressources doit être allouée et où sur l'infrastructure elles doivent être déployées pour qu'elles soient conformes aux licences tierces.
- **Environnements cohérents** OpenShift s'assure que l'environnement mis à disposition des développeurs et tout au long du cycle de vie de l'application est cohérent, du système d'exploitation aux bibliothèques, en passant par la version d'exécution (par exemple, Java Runtime) et même l'exécution de l'application en cours d'utilisation (par exemple, Tomcat) afin d'éliminer les risques provenant d'environnements incohérents.
- **Gestion de la configuration** La gestion de la configuration et des données sensibles est intégrée à la plateforme pour garantir qu'une configuration d'application cohérente et indépendante de l'environnement est fournie à l'application, quelles que soient les technologies utilisées pour créer l'application ou l'environnement dans lequel elle est déployée.

- **Journaux d'application et métriques.** Un retour d'information rapide est un aspect important du développement d'applications. La surveillance intégrée et la gestion des journaux d'OpenShift fournissent des mesures immédiates aux développeurs afin qu'ils puissent étudier le comportement de l'application à travers les changements et être en mesure de résoudre les problèmes le plus tôt possible dans le cycle de vie de l'application.
- **Catalogue de sécurité et de conteneurs** OpenShift offre une multilocation et protège l'utilisateur contre l'exécution de code nuisible en utilisant une sécurité établie avec Security-Enhanced Linux (SELinux), CGroups et Secure Computing Mode (seccomp) pour isoler et protéger les conteneurs. Il fournit également un cryptage via des certificats TLS pour les différents sous-systèmes et un accès aux conteneurs certifiés Red Hat (access.redhat.com/containers) qui sont analysés et classés avec un accent particulier sur la sécurité pour fournir des conteneurs d'applications certifiés, fiables et sécurisés aux utilisateurs finaux.



Méthodes de déploiement pour Red Hat OpenShift

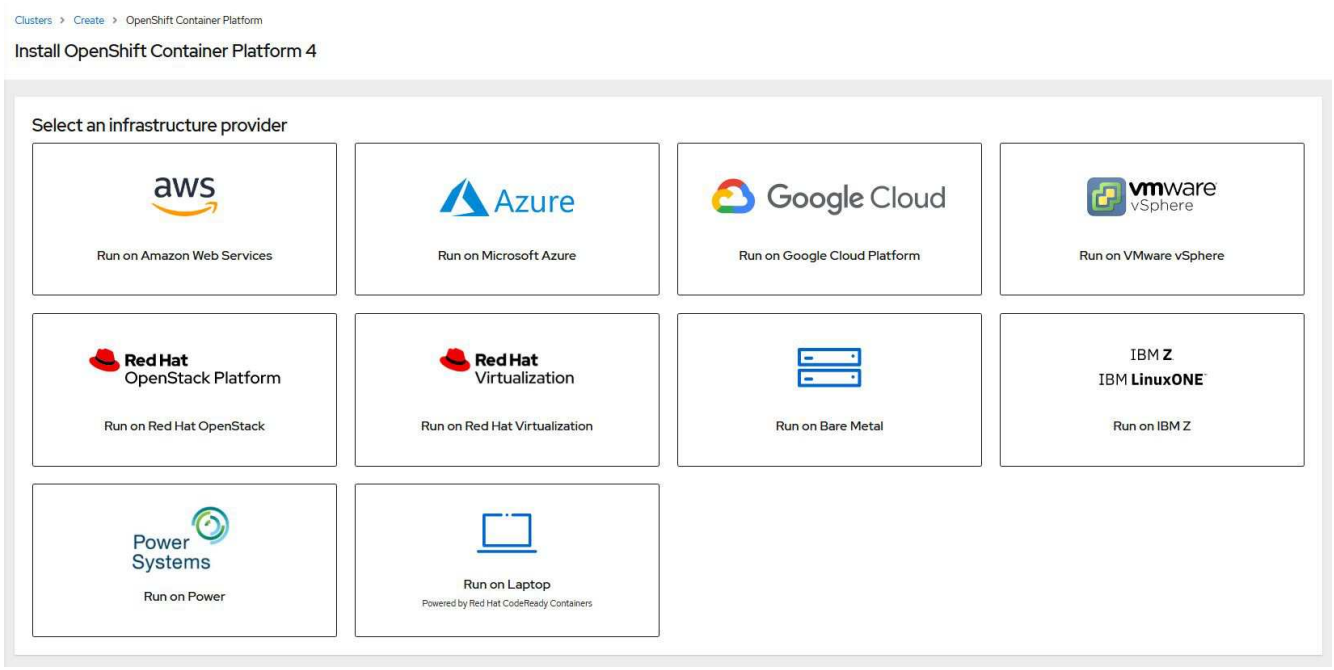
À partir de Red Hat OpenShift 4, les méthodes de déploiement pour OpenShift incluent des déploiements manuels à l'aide de l'infrastructure provisionnée par l'utilisateur (UPI) pour des déploiements hautement personnalisés ou des déploiements entièrement automatisés à l'aide de l'infrastructure provisionnée par l'installateur (IPI).

La méthode d'installation IPI est la méthode préférée dans la plupart des cas car elle permet le déploiement rapide de clusters OpenShift pour les environnements de développement, de test et de production.

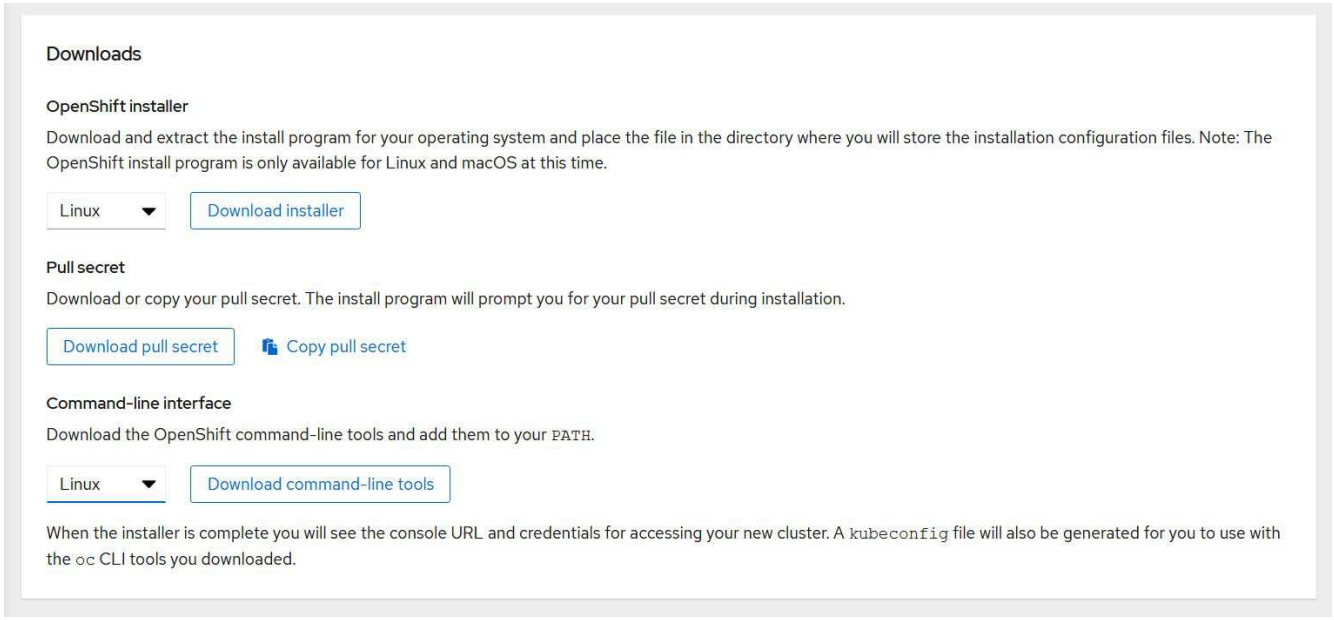
Installation IPI de Red Hat OpenShift

Le déploiement de l'infrastructure provisionnée par l'installateur (IPI) d'OpenShift implique les étapes de haut niveau suivantes :

1. Visitez Red Hat OpenShift"site web" et connectez-vous avec vos identifiants SSO.
2. Sélectionnez l'environnement dans lequel vous souhaitez déployer Red Hat OpenShift.



3. Sur l'écran suivant, téléchargez le programme d'installation, le secret d'extraction unique et les outils CLI pour la gestion.



4. Suivez le"instructions d'installation" fourni par Red Hat pour être déployé dans l'environnement de votre choix.

Déploiements OpenShift validés par NetApp

NetApp a testé et validé le déploiement de Red Hat OpenShift dans ses laboratoires à l'aide de la méthode de déploiement Installer Provisioned Infrastructure (IPI) dans chacun des environnements de centre de données suivants :

- "OpenShift sur Bare Metal"
- "OpenShift sur la plateforme Red Hat OpenStack"

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

OpenShift sur Bare Metal

OpenShift on Bare Metal fournit un déploiement automatisé de la plateforme de conteneurs OpenShift sur des serveurs standard.

OpenShift sur Bare Metal est similaire aux déploiements virtuels d'OpenShift, qui offrent une facilité de déploiement, un provisionnement rapide et une mise à l'échelle des clusters OpenShift, tout en prenant en charge les charges de travail virtualisées pour les applications qui ne sont pas prêtes à être conteneurisées. En déployant sur du bare metal, vous n'avez pas besoin de la surcharge supplémentaire nécessaire pour gérer l'environnement de l'hyperviseur hôte en plus de l'environnement OpenShift. En déployant directement sur des serveurs bare metal, vous pouvez également réduire les limitations de surcharge physique liées au partage des ressources entre l'hôte et l'environnement OpenShift.

OpenShift sur Bare Metal offre les fonctionnalités suivantes :

- **IPI ou déploiement d'installateur assisté** Avec un cluster OpenShift déployé par Installer Provisioned Infrastructure (IPI) sur des serveurs bare metal, les clients peuvent déployer un environnement OpenShift hautement polyvalent et facilement évolutif directement sur des serveurs standard, sans avoir besoin de gérer une couche d'hyperviseur.
- **Conception de cluster compacte** Pour minimiser les exigences matérielles, OpenShift sur bare metal permet aux utilisateurs de déployer des clusters de seulement 3 nœuds, en permettant aux nœuds du plan de contrôle OpenShift d'agir également comme nœuds de travail et conteneurs hôtes.
- **Virtualisation OpenShift** OpenShift peut exécuter des machines virtuelles dans des conteneurs en utilisant OpenShift Virtualization. Cette virtualisation native du conteneur exécute l'hyperviseur KVM à l'intérieur d'un conteneur et attache des volumes persistants pour le stockage de la machine virtuelle.
- **Infrastructure optimisée pour l'IA/ML** Déployez des applications telles que Kubeflow pour les applications d'apprentissage automatique en incorporant des nœuds de travail basés sur GPU à votre environnement OpenShift et en tirant parti de la planification avancée d'OpenShift.

Conception de réseau

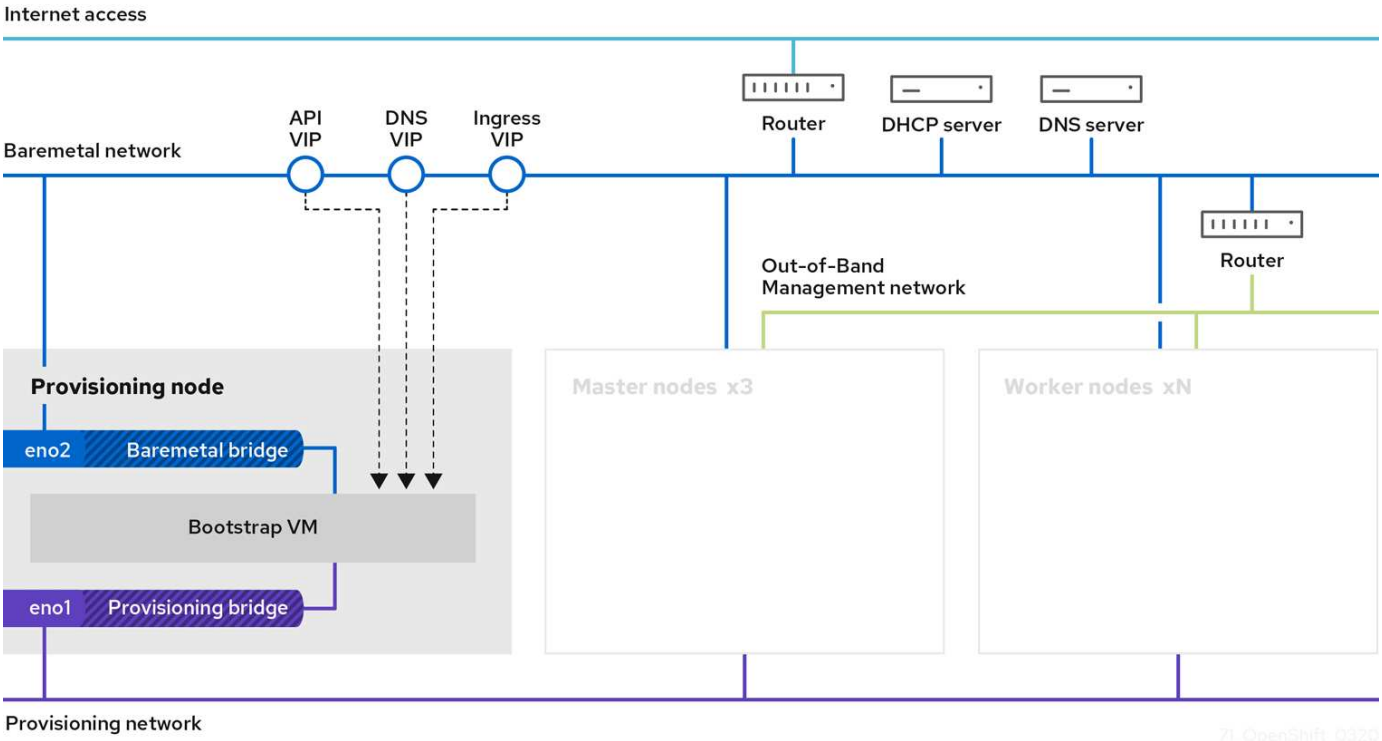
La solution Red Hat OpenShift sur NetApp utilise deux commutateurs de données pour fournir une connectivité de données principale à 25 Gbit/s. Il utilise également deux commutateurs de gestion qui fournissent une connectivité à 1 Gbit/s pour la gestion en bande des nœuds de stockage et la gestion hors bande pour la fonctionnalité IPMI.

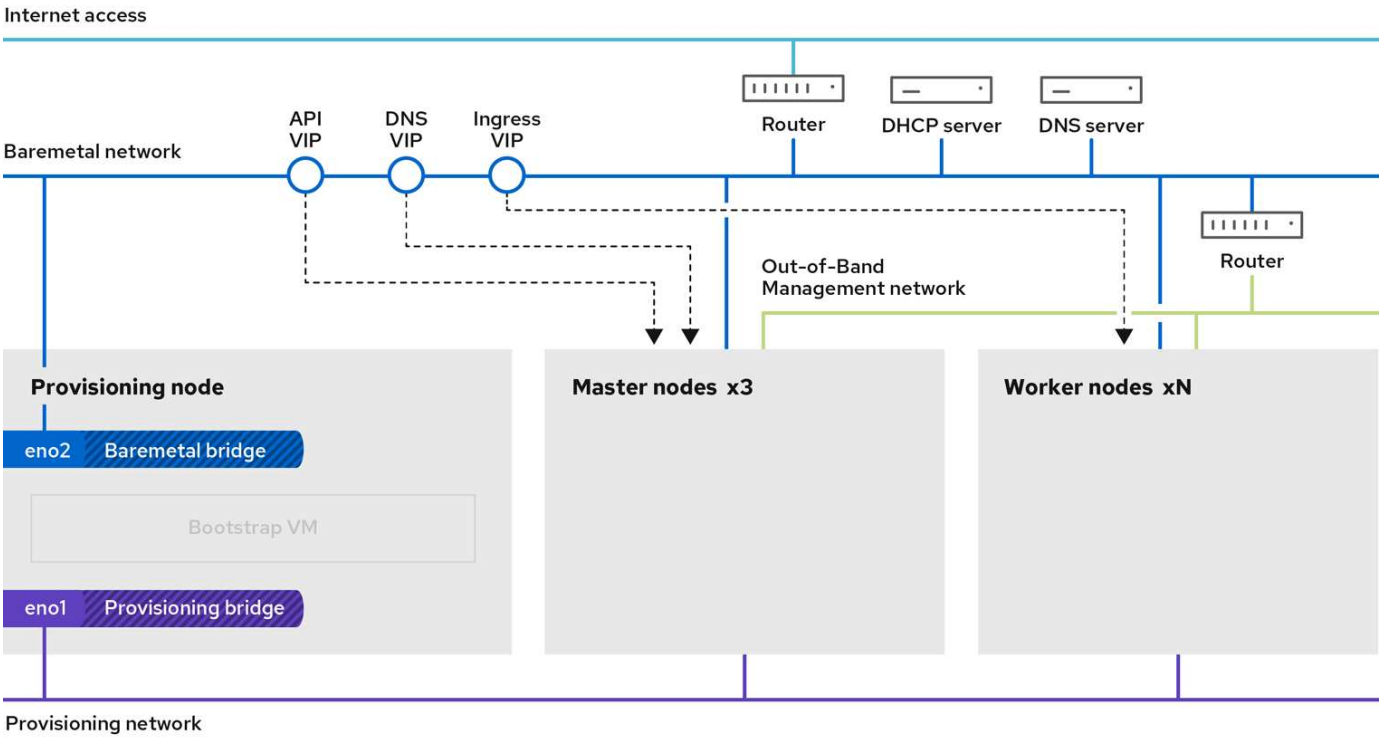
Pour le déploiement IPI bare-metal d'OpenShift, vous devez créer un nœud de provisionnement, une machine Red Hat Enterprise Linux 8 qui doit avoir des interfaces réseau connectées à des réseaux distincts.

- **Réseau de provisionnement** Ce réseau est utilisé pour démarrer les nœuds bare-metal et installer les images et packages nécessaires pour déployer le cluster OpenShift.
- **Réseau bare-metal** Ce réseau est utilisé pour la communication publique du cluster après son déploiement.

Pour la configuration du nœud de provisionnement, le client crée des interfaces de pont qui permettent au trafic d'être acheminé correctement sur le nœud lui-même et sur la machine virtuelle Bootstrap provisionnée à des fins de déploiement. Une fois le cluster déployé, l'API et les adresses VIP d'entrée sont migrées du nœud d'amorçage vers le cluster nouvellement déployé.

Les images suivantes illustrent l'environnement pendant le déploiement de l'IPI et une fois le déploiement terminé.





Exigences VLAN

La solution Red Hat OpenShift avec NetApp est conçue pour séparer logiquement le trafic réseau à différentes fins en utilisant des réseaux locaux virtuels (VLAN).

VLAN	But	ID VLAN
Réseau de gestion hors bande	Gestion des nœuds bare metal et IPMI	16
Réseau bare metal	Réseau pour les services OpenShift une fois le cluster disponible	181
Réseau d'approvisionnement	Réseau pour le démarrage PXE et l'installation de nœuds bare metal via IPI	3485

Bien que chacun de ces réseaux soit virtuellement séparé par des VLAN, chaque port physique doit être configuré en mode d'accès avec le VLAN principal attribué, car il n'existe aucun moyen de transmettre une balise VLAN pendant une séquence de démarrage PXE.

Ressources de soutien à l'infrastructure réseau

L'infrastructure suivante doit être en place avant le déploiement de la plateforme de conteneurs OpenShift :

- Au moins un serveur DNS qui fournit une résolution complète du nom d'hôte accessible depuis le réseau de gestion en bande et le réseau VM.
- Au moins un serveur NTP accessible depuis le réseau de gestion en bande et le réseau VM.
- (Facultatif) Connectivité Internet sortante pour le réseau de gestion en bande et le réseau VM.

OpenShift sur la plateforme Red Hat OpenStack

La plateforme Red Hat OpenStack offre une base intégrée pour créer, déployer et faire évoluer un cloud OpenStack privé sécurisé et fiable.

OSP est un cloud d'infrastructure en tant que service (IaaS) mis en œuvre par un ensemble de services de contrôle qui gèrent les ressources de calcul, de stockage et de réseau. L'environnement est géré à l'aide d'une interface Web qui permet aux administrateurs et aux utilisateurs de contrôler, de provisionner et d'automatiser les ressources OpenStack. De plus, l'infrastructure OpenStack est facilitée par une interface de ligne de commande étendue et une API permettant des capacités d'automatisation complètes pour les administrateurs et les utilisateurs finaux.

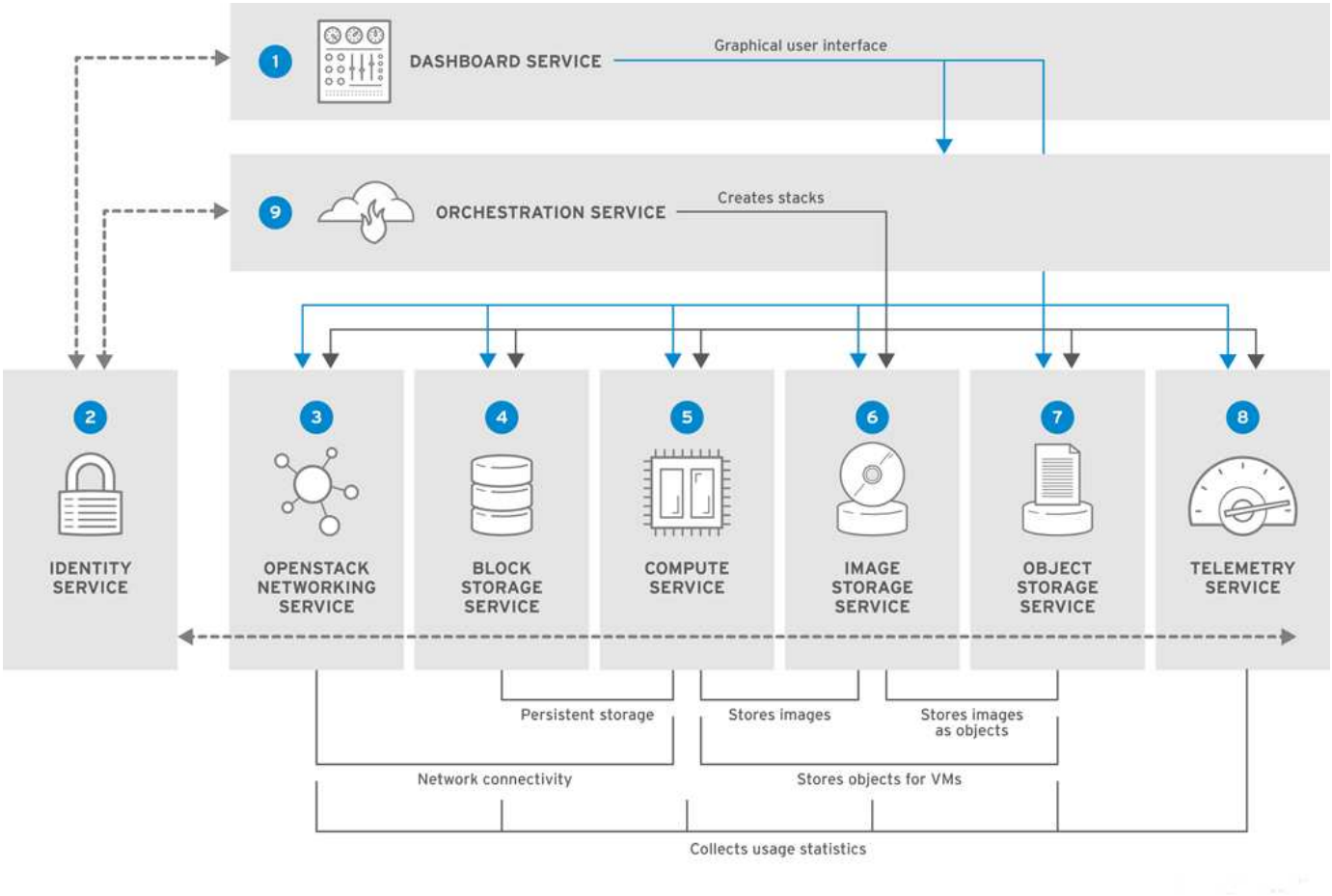
Le projet OpenStack est un projet communautaire développé rapidement qui fournit des versions mises à jour tous les six mois. Au départ, Red Hat OpenStack Platform a suivi le rythme de ce cycle de publication en publiant une nouvelle version avec chaque version en amont et en fournissant un support à long terme pour chaque troisième version. Récemment, avec la sortie d'OSP 16.0 (basée sur OpenStack Train), Red Hat a choisi de ne pas suivre le rythme des numéros de version, mais a plutôt rétroporté de nouvelles fonctionnalités dans des sous-versions. La version la plus récente est Red Hat OpenStack Platform 16.1, qui inclut des fonctionnalités avancées rétroportées des versions Ussuri et Victoria en amont.

Pour plus d'informations sur OSP, consultez le"Site Web de la plateforme Red Hat OpenStack" .

Services OpenStack

Les services de la plateforme OpenStack sont déployés sous forme de conteneurs, ce qui isole les services les uns des autres et permet des mises à niveau faciles. La plateforme OpenStack utilise un ensemble de

conteneurs construits et gérés avec Kolla. Le déploiement des services est effectué en extrayant des images de conteneurs du portail personnalisé Red Hat. Ces conteneurs de services sont gérés à l'aide de la commande Podman et sont déployés, configurés et maintenus avec Red Hat OpenStack Director.



Service	Nom du projet	Description
Tableau de bord	Horizon	Tableau de bord basé sur un navigateur Web que vous utilisez pour gérer les services OpenStack.
Identité	Keystone	Service centralisé pour l'authentification et l'autorisation des services OpenStack et pour la gestion des utilisateurs, des projets et des rôles.
Réseau OpenStack	Neutron	Fournit la connectivité entre les interfaces des services OpenStack.
Stockage en bloc	Cendre	Gère les volumes de stockage de blocs persistants pour les machines virtuelles (VM).
Calculer	Nova	Gère et provisionne les machines virtuelles exécutées sur les nœuds de calcul.
Image	Coup d'oeil	Service de registre utilisé pour stocker des ressources telles que des images de machine virtuelle et des instantanés de volume.
Stockage d'objets	Rapide	Permet aux utilisateurs de stocker et de récupérer des fichiers et des données arbitraires.
Télémétrie	Célomètre	Fournit des mesures de l'utilisation des ressources cloud.

Service	Nom du projet	Description
Orchestration	Chaleur	Moteur d'orchestration basé sur des modèles qui prend en charge la création automatique de piles de ressources.

Conception de réseau

La solution Red Hat OpenShift avec NetApp utilise deux commutateurs de données pour fournir une connectivité de données principale à 25 Gbit/s. Il utilise également deux commutateurs de gestion supplémentaires qui fournissent une connectivité à 1 Gbit/s pour la gestion en bande des nœuds de stockage et la gestion hors bande pour la fonctionnalité IPMI.

La fonctionnalité IPMI est requise par Red Hat OpenStack Director pour déployer Red Hat OpenStack Platform à l'aide du service de provisionnement bare-metal Ironic.

Exigences VLAN

Red Hat OpenShift avec NetApp est conçu pour séparer logiquement le trafic réseau à des fins différentes en utilisant des réseaux locaux virtuels (VLAN). Cette configuration peut être mise à l'échelle pour répondre aux demandes des clients ou pour fournir une isolation supplémentaire pour des services réseau spécifiques. Le tableau suivant répertorie les VLAN requis pour implémenter la solution lors de la validation de la solution chez NetApp.

VLAN	But	ID VLAN
Réseau de gestion hors bande	Réseau utilisé pour la gestion des nœuds physiques et du service IPMI pour Ironic.	16
Infrastructure de stockage	Réseau utilisé pour les nœuds de contrôleur pour mapper les volumes directement afin de prendre en charge les services d'infrastructure comme Swift.	201
Cendres de stockage	Réseau utilisé pour mapper et attacher des volumes de blocs directement aux instances virtuelles déployées dans l'environnement.	202
API interne	Réseau utilisé pour la communication entre les services OpenStack à l'aide de la communication API, des messages RPC et de la communication de base de données.	301
Locataire	Neutron fournit à chaque locataire ses propres réseaux via le tunneling via VXLAN. Le trafic réseau est isolé au sein de chaque réseau locataire. Chaque réseau locataire possède un sous-réseau IP qui lui est associé, et les espaces de noms réseau signifient que plusieurs réseaux locataires peuvent utiliser la même plage d'adresses sans provoquer de conflits.	302
Gestion du stockage	OpenStack Object Storage (Swift) utilise ce réseau pour synchroniser les objets de données entre les nœuds de réplica participants. Le service proxy agit comme interface intermédiaire entre les demandes des utilisateurs et la couche de stockage sous-jacente. Le proxy reçoit les demandes entrantes et localise la réplique nécessaire pour récupérer les données demandées.	303
PXE	OpenStack Director fournit un démarrage PXE dans le cadre du service de provisionnement bare metal d'Ironic pour orchestrer l'installation d'OSP Overcloud.	3484

VLAN	But	ID VLAN
Externe	Réseau accessible au public qui héberge le tableau de bord OpenStack (Horizon) pour la gestion graphique et permet des appels d'API publics pour gérer les services OpenStack.	3485
Réseau de gestion en bande	Fournit l'accès aux fonctions d'administration système telles que l'accès SSH, le trafic DNS et le trafic Network Time Protocol (NTP). Ce réseau agit également comme une passerelle pour les nœuds non contrôleurs.	3486

Ressources de soutien à l'infrastructure réseau

L'infrastructure suivante doit être en place avant le déploiement de la plateforme de conteneurs OpenShift :

- Au moins un serveur DNS qui fournit une résolution complète du nom d'hôte.
- Au moins trois serveurs NTP qui peuvent maintenir la synchronisation horaire des serveurs de la solution.
- (Facultatif) Connectivité Internet sortante pour l'environnement OpenShift.

Bonnes pratiques pour les déploiements de production

Cette section répertorie plusieurs bonnes pratiques qu'une organisation doit prendre en compte avant de déployer cette solution en production.

Déployer OpenShift sur un cloud privé OSP avec au moins trois nœuds de calcul

L'architecture vérifiée décrite dans ce document présente le déploiement matériel minimal adapté aux opérations HA en déployant trois nœuds de contrôleur OSP et deux nœuds de calcul OSP. Cette architecture garantit une configuration tolérante aux pannes dans laquelle les deux nœuds de calcul peuvent lancer des instances virtuelles et les machines virtuelles déployées peuvent migrer entre les deux hyperviseurs.

Étant donné que Red Hat OpenShift se déploie initialement avec trois nœuds maîtres, une configuration à deux nœuds peut entraîner l'occupation du même nœud par au moins deux maîtres, ce qui peut entraîner une éventuelle panne d'OpenShift si ce nœud spécifique devient indisponible. Par conséquent, il s'agit d'une bonne pratique de Red Hat de déployer au moins trois nœuds de calcul OSP afin que les maîtres OpenShift puissent être répartis uniformément et que la solution reçoive un degré supplémentaire de tolérance aux pannes.

Configurer l'affinité machine virtuelle/hôte

La distribution des maîtres OpenShift sur plusieurs nœuds d'hyperviseur peut être réalisée en activant l'affinité VM/hôte.

L'affinité est un moyen de définir des règles pour un ensemble de machines virtuelles et/ou d'hôtes qui déterminent si les machines virtuelles s'exécutent ensemble sur le même hôte ou sur des hôtes du groupe ou sur des hôtes différents. Il est appliqué aux machines virtuelles en créant des groupes d'affinité constitués de machines virtuelles et/ou d'hôtes avec un ensemble de paramètres et de conditions identiques. Selon que les machines virtuelles d'un groupe d'affinité s'exécutent sur le même hôte ou sur les mêmes hôtes du groupe ou séparément sur des hôtes différents, les paramètres du groupe d'affinité peuvent définir une affinité positive ou une affinité négative. Dans la plateforme Red Hat OpenStack, les règles d'affinité et d'anti-affinité d'hôte peuvent être créées et appliquées en créant des groupes de serveurs et en configurant des filtres afin que les instances déployées par Nova dans un groupe de serveurs se déploient sur différents nœuds de calcul.

Un groupe de serveurs dispose d'un maximum par défaut de 10 instances virtuelles dont il peut gérer le placement. Cela peut être modifié en mettant à jour les quotas par défaut pour Nova.

Il existe une limite d'affinité/anti-affinité spécifique pour les groupes de serveurs OSP ; s'il n'y a pas suffisamment de ressources à déployer sur des nœuds séparés ou pas suffisamment de ressources pour permettre le partage de nœuds, la machine virtuelle ne parvient pas à démarrer.

Pour configurer les groupes d'affinité, voir"Comment configurer Affinity et Anti-Affinity pour les instances OpenStack ?" .

Utiliser un fichier d'installation personnalisé pour le déploiement d'OpenShift

IPI facilite le déploiement des clusters OpenShift grâce à l'assistant interactif décrit précédemment dans ce document. Cependant, il est possible que vous ayez besoin de modifier certaines valeurs par défaut dans le cadre d'un déploiement de cluster.

Dans ces cas-là, vous pouvez exécuter l'assistant et configurer les tâches sans déployer immédiatement un cluster ; celui-ci crée plutôt un fichier de configuration à partir duquel le cluster pourra être déployé ultérieurement. Ceci est très utile si vous devez modifier les paramètres par défaut d'IPI, ou si vous souhaitez déployer plusieurs clusters identiques dans votre environnement pour d'autres usages tels que la mutualisation. Pour plus d'informations sur la création d'une configuration d'installation personnalisée pour OpenShift, consultez "Red Hat OpenShift Installation d'un cluster sur OpenStack avec personnalisations".

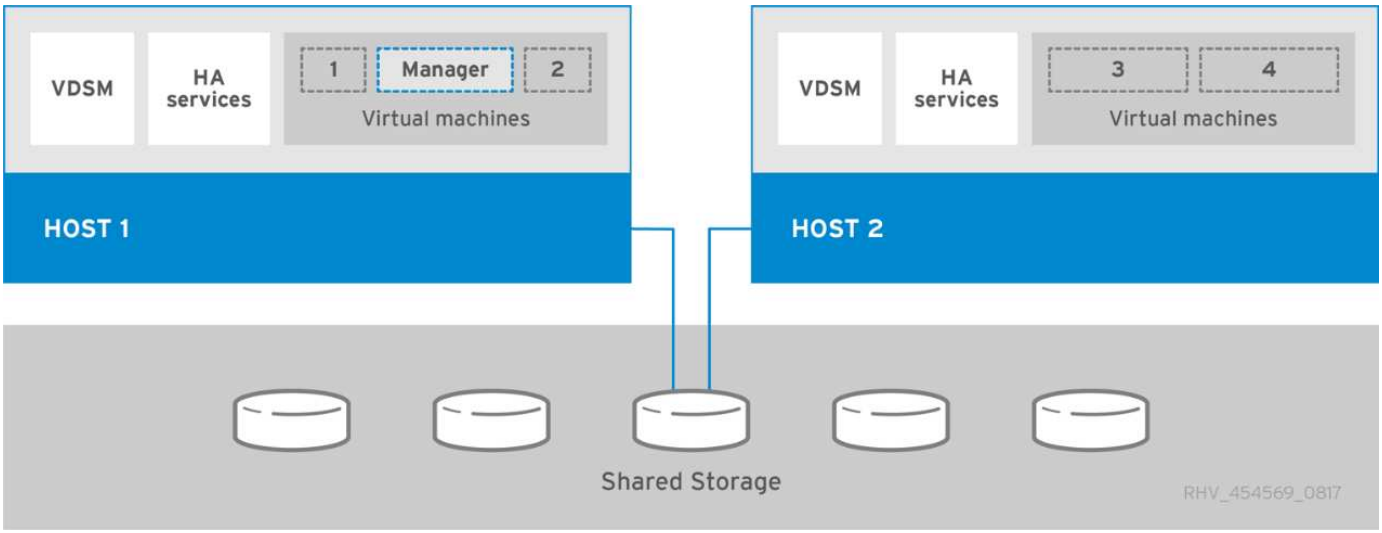
OpenShift sur Red Hat Virtualization

Red Hat Virtualization (RHV) est une plate-forme de centre de données virtuel d'entreprise qui s'exécute sur Red Hat Enterprise Linux (RHEL) et utilise l'hyperviseur KVM.

Pour plus d'informations sur RHV, consultez le"Site Web de virtualisation Red Hat" .

RHV offre les fonctionnalités suivantes :

- **Gestion centralisée des machines virtuelles et des hôtes** Le gestionnaire RHV s'exécute comme une machine physique ou virtuelle (VM) dans le déploiement et fournit une interface graphique Web pour la gestion de la solution à partir d'une interface centrale.
- **Moteur auto-hébergé** Pour minimiser les exigences matérielles, RHV permet à RHV Manager (RHV-M) d'être déployé en tant que machine virtuelle sur les mêmes hôtes qui exécutent les machines virtuelles invitées.
- **Haute disponibilité** Pour éviter toute interruption en cas de panne de l'hôte, RHV permet de configurer les machines virtuelles pour une haute disponibilité. Les machines virtuelles hautement disponibles sont contrôlées au niveau du cluster à l'aide de politiques de résilience.
- **Haute évolutivité** Un seul cluster RHV peut avoir jusqu'à 200 hôtes hyperviseurs, ce qui lui permet de prendre en charge les exigences des machines virtuelles massives pour héberger des charges de travail gourmandes en ressources et de niveau entreprise.
- **Sécurité renforcée** Héritées de RHV, les technologies Secure Virtualization (sVirt) et Security Enhanced Linux (SELinux) sont utilisées par RHV à des fins de sécurité renforcée et de renforcement des hôtes et des machines virtuelles. Le principal avantage de ces fonctionnalités est l'isolement logique d'une machine virtuelle et de ses ressources associées.



Conception de réseau

La solution Red Hat OpenShift sur NetApp utilise deux commutateurs de données pour fournir une connectivité de données principale à 25 Gbit/s. Il utilise également deux commutateurs de gestion supplémentaires qui fournissent une connectivité à 1 Gbit/s pour la gestion en bande des nœuds de stockage et la gestion hors bande pour la fonctionnalité IPMI. OCP utilise le réseau logique de la machine virtuelle sur RHV pour la gestion des clusters. Cette section décrit la disposition et l'objectif de chaque segment de réseau virtuel utilisé dans la solution et décrit les conditions préalables au déploiement de la solution.

Exigences VLAN

Red Hat OpenShift sur RHV est conçu pour séparer logiquement le trafic réseau à des fins différentes en utilisant des réseaux locaux virtuels (VLAN). Cette configuration peut être mise à l'échelle pour répondre aux demandes des clients ou pour fournir une isolation supplémentaire pour des services réseau spécifiques. Le tableau suivant répertorie les VLAN requis pour implémenter la solution lors de la validation de la solution chez NetApp.

VLAN	But	ID VLAN
Réseau de gestion hors bande	Gestion des nœuds physiques et IPMI	16
Réseau VM	Accès réseau invité virtuel	1172
Réseau de gestion en bande	Gestion des nœuds RHV-H, RHV-Manager et réseau ovirtmgmt	3343
Réseau de stockage	Réseau de stockage pour NetApp Element iSCSI	3344
Réseau de migration	Réseau pour la migration des invités virtuels	3345

Ressources de soutien à l'infrastructure réseau

L'infrastructure suivante doit être en place avant le déploiement de la plateforme de conteneurs OpenShift :

- Au moins un serveur DNS fournissant une résolution complète du nom d'hôte accessible depuis le réseau de gestion en bande et le réseau VM.
- Au moins un serveur NTP accessible depuis le réseau de gestion en bande et le réseau VM.

- (Facultatif) Connectivité Internet sortante pour le réseau de gestion en bande et le réseau VM.

Bonnes pratiques pour les déploiements de production

Cette section répertorie plusieurs bonnes pratiques qu'une organisation doit prendre en compte avant de déployer cette solution en production.

Déployer OpenShift sur un cluster RHV d'au moins trois nœuds

L'architecture vérifiée décrite dans ce document présente le déploiement matériel minimal adapté aux opérations HA en déployant deux nœuds d'hyperviseur RHV-H et en garantissant une configuration tolérante aux pannes où les deux hôtes peuvent gérer le moteur hébergé et les machines virtuelles déployées peuvent migrer entre les deux hyperviseurs.

Étant donné que Red Hat OpenShift se déploie initialement avec trois nœuds maîtres, il est garanti dans une configuration à deux nœuds qu'au moins deux maîtres occuperont le même nœud, ce qui peut entraîner une éventuelle panne d'OpenShift si ce nœud spécifique devient indisponible. Par conséquent, il s'agit d'une bonne pratique de Red Hat qu'au moins trois nœuds d'hyperviseur RHV-H soient déployés dans le cadre de la solution afin que les maîtres OpenShift puissent être répartis uniformément et que la solution reçoive un degré supplémentaire de tolérance aux pannes.

Configurer l'affinité machine virtuelle/hôte

Vous pouvez distribuer les maîtres OpenShift sur plusieurs nœuds d'hyperviseur en activant l'affinité VM/hôte.

L'affinité est un moyen de définir des règles pour un ensemble de machines virtuelles et/ou d'hôtes qui déterminent si les machines virtuelles s'exécutent ensemble sur le même hôte ou sur des hôtes du groupe ou sur des hôtes différents. Il est appliqué aux machines virtuelles en créant des groupes d'affinité constitués de machines virtuelles et/ou d'hôtes avec un ensemble de paramètres et de conditions identiques. Selon que les machines virtuelles d'un groupe d'affinité s'exécutent sur le même hôte ou sur les mêmes hôtes du groupe ou séparément sur des hôtes différents, les paramètres du groupe d'affinité peuvent définir une affinité positive ou une affinité négative.

Les conditions définies pour les paramètres peuvent être soit strictes, soit souples. Une application stricte garantit que les machines virtuelles d'un groupe d'affinité suivent toujours l'affinité positive ou négative de manière stricte, sans tenir compte des conditions externes. L'application souple garantit qu'une préférence plus élevée est définie pour les machines virtuelles d'un groupe d'affinité afin de suivre l'affinité positive ou négative chaque fois que cela est possible. Dans la configuration à deux ou trois hyperviseurs décrite dans ce document, l'affinité douce est le paramètre recommandé. Dans les clusters plus grands, l'affinité dure peut distribuer correctement les nœuds OpenShift.

Pour configurer les groupes d'affinité, consultez le"Red Hat 6.11. Documentation sur les groupes d'affinité" .

Utiliser un fichier d'installation personnalisé pour le déploiement d'OpenShift

IPI facilite le déploiement des clusters OpenShift grâce à l'assistant interactif décrit précédemment dans ce document. Cependant, il est possible que certaines valeurs par défaut doivent être modifiées dans le cadre du déploiement du cluster.

Dans ces cas, vous pouvez exécuter l'assistant sans déployer immédiatement un cluster. Au lieu de cela, un fichier de configuration est créé à partir duquel le cluster peut être déployé ultérieurement. Ceci est très utile si vous souhaitez modifier les valeurs par défaut de l'IPI ou si vous souhaitez déployer plusieurs clusters identiques dans votre environnement pour d'autres utilisations telles que le multi-hébergement. Pour plus d'informations sur la création d'une configuration d'installation personnalisée pour OpenShift, consultez"Red Hat OpenShift Installation d'un cluster sur RHV avec personnalisations" .

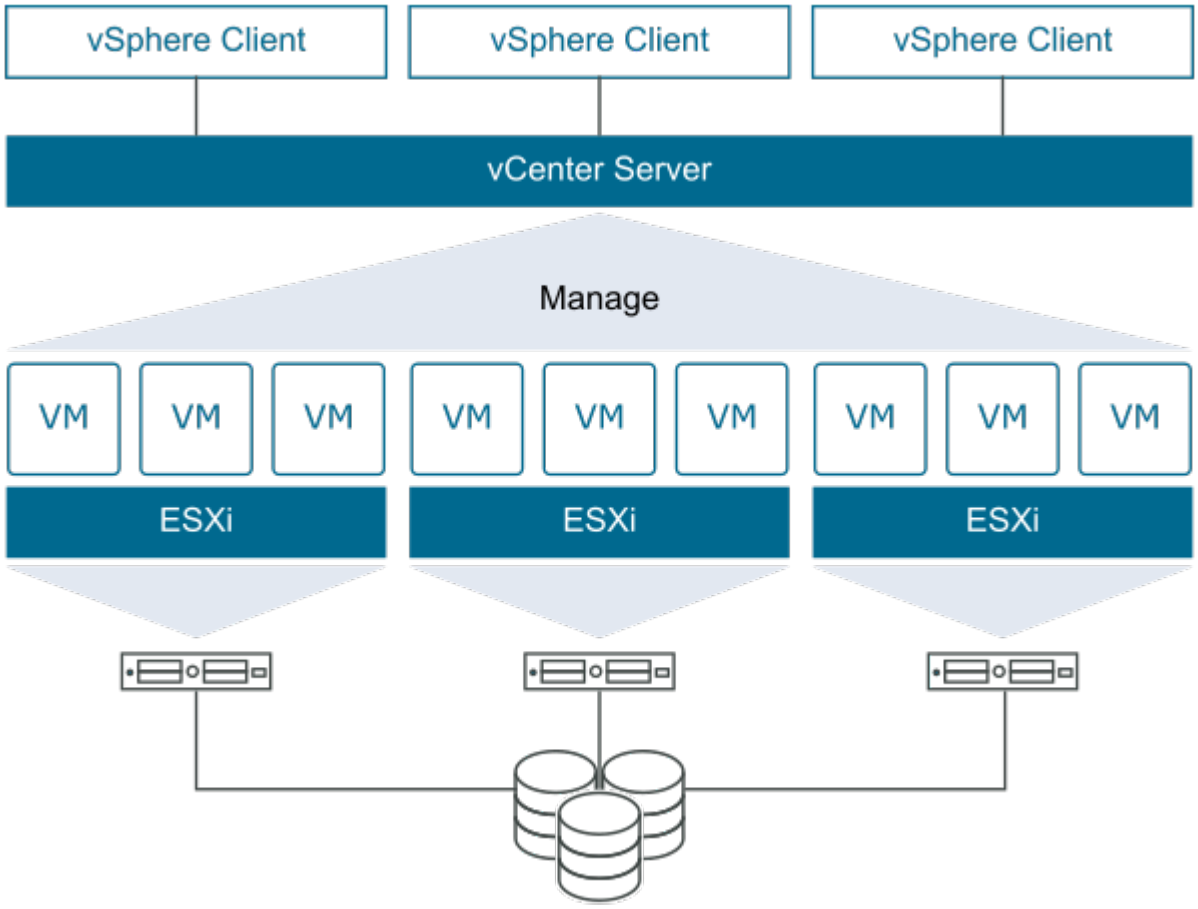
OpenShift sur VMware vSphere

VMware vSphere est une plate-forme de virtualisation permettant de gérer de manière centralisée un grand nombre de serveurs et de réseaux virtualisés exécutés sur l'hyperviseur ESXi.

Pour plus d'informations sur VMware vSphere, consultez le"Site Web de VMware vSphere" .

VMware vSphere offre les fonctionnalités suivantes :

- **VMware vCenter Server** VMware vCenter Server fournit une gestion unifiée de tous les hôtes et machines virtuelles à partir d'une console unique et regroupe la surveillance des performances des clusters, des hôtes et des machines virtuelles.
- **VMware vSphere vMotion** VMware vCenter vous permet de migrer à chaud des machines virtuelles entre les nœuds du cluster sur demande et sans interruption.
- **vSphere Haute Disponibilité** Pour éviter toute interruption en cas de panne de l'hôte, VMware vSphere permet de regrouper les hôtes en cluster et de les configurer pour une haute disponibilité. Les machines virtuelles perturbées par une panne d'hôte sont redémarrées rapidement sur d'autres hôtes du cluster, restaurant ainsi les services.
- **Planificateur de ressources distribuées (DRS)** Un cluster VMware vSphere peut être configuré pour équilibrer la charge des besoins en ressources des machines virtuelles qu'il héberge. Les machines virtuelles présentant des conflits de ressources peuvent être migrées à chaud vers d'autres nœuds du cluster pour garantir que suffisamment de ressources sont disponibles.



Conception de réseau

La solution Red Hat OpenShift sur NetApp utilise deux commutateurs de données pour fournir une connectivité de données principale à 25 Gbit/s. Il utilise également deux commutateurs de gestion supplémentaires qui fournissent une connectivité à 1 Gbit/s pour la gestion en bande des nœuds de stockage et la gestion hors bande pour la fonctionnalité IPMI. OCP utilise le réseau logique VM sur VMware vSphere pour la gestion de ses clusters. Cette section décrit la disposition et l'objectif de chaque segment de réseau virtuel utilisé dans la solution et décrit les conditions préalables au déploiement de la solution.

Exigences VLAN

Red Hat OpenShift sur VMware vSphere est conçu pour séparer logiquement le trafic réseau à des fins différentes en utilisant des réseaux locaux virtuels (VLAN). Cette configuration peut être mise à l'échelle pour répondre aux demandes des clients ou pour fournir une isolation supplémentaire pour des services réseau spécifiques. Le tableau suivant répertorie les VLAN requis pour implémenter la solution lors de la validation de la solution chez NetApp.

VLAN	But	ID VLAN
Réseau de gestion hors bande	Gestion des nœuds physiques et IPMI	16
Réseau VM	Accès réseau invité virtuel	181
Réseau de stockage	Réseau de stockage pour ONTAP NFS	184
Réseau de stockage	Réseau de stockage pour ONTAP iSCSI	185
Réseau de gestion en bande	Gestion des nœuds ESXi, VCenter Server, ONTAP Select	3480
Réseau de stockage	Réseau de stockage pour NetApp Element iSCSI	3481
Réseau de migration	Réseau pour la migration des invités virtuels	3482

Ressources de soutien à l'infrastructure réseau

L'infrastructure suivante doit être en place avant le déploiement de la plateforme de conteneurs OpenShift :

- Au moins un serveur DNS fournissant une résolution complète du nom d'hôte accessible depuis le réseau de gestion en bande et le réseau VM.
- Au moins un serveur NTP accessible depuis le réseau de gestion en bande et le réseau VM.
- (Facultatif) Connectivité Internet sortante pour le réseau de gestion en bande et le réseau VM.

Bonnes pratiques pour les déploiements de production

Cette section répertorie plusieurs bonnes pratiques qu'une organisation doit prendre en compte avant de déployer cette solution en production.

Déployer OpenShift sur un cluster ESXi d'au moins trois nœuds

L'architecture vérifiée décrite dans ce document présente le déploiement matériel minimal adapté aux opérations HA en déployant deux nœuds d'hyperviseur ESXi et en garantissant une configuration tolérante aux pannes en activant VMware vSphere HA et VMware vMotion. Cette configuration permet aux machines virtuelles déployées de migrer entre les deux hyperviseurs et de redémarrer si un hôte devient indisponible.

Étant donné que Red Hat OpenShift se déploie initialement avec trois nœuds maîtres, au moins deux maîtres dans une configuration à deux nœuds peuvent occuper le même nœud dans certaines circonstances, ce qui peut entraîner une éventuelle panne d'OpenShift si ce nœud spécifique devient indisponible. Par conséquent, il s'agit d'une bonne pratique de Red Hat qu'au moins trois nœuds d'hyperviseur ESXi doivent être déployés afin que les maîtres OpenShift puissent être répartis uniformément, ce qui offre un degré supplémentaire de tolérance aux pannes.

Configurer l'affinité de la machine virtuelle et de l'hôte

Il est possible de garantir la distribution des maîtres OpenShift sur plusieurs nœuds d'hyperviseur en activant l'affinité entre la machine virtuelle et l'hôte.

L'affinité ou l'anti-affinité est un moyen de définir des règles pour un ensemble de machines virtuelles et/ou d'hôtes qui déterminent si les machines virtuelles s'exécutent ensemble sur le même hôte ou sur des hôtes du groupe ou sur des hôtes différents. Il est appliqué aux machines virtuelles en créant des groupes d'affinité constitués de machines virtuelles et/ou d'hôtes avec un ensemble de paramètres et de conditions identiques. Selon que les machines virtuelles d'un groupe d'affinité s'exécutent sur le même hôte ou sur les mêmes hôtes du groupe ou séparément sur des hôtes différents, les paramètres du groupe d'affinité peuvent définir une affinité positive ou une affinité négative.

Pour configurer les groupes d'affinité, consultez la section "Documentation vSphere 9.0 : Utilisation des règles d'affinité DRS".

Utiliser un fichier d'installation personnalisé pour le déploiement d'OpenShift

IPI facilite le déploiement des clusters OpenShift grâce à l'assistant interactif décrit précédemment dans ce document. Cependant, il est possible que vous ayez besoin de modifier certaines valeurs par défaut dans le cadre d'un déploiement de cluster.

Dans ces cas, vous pouvez exécuter l'assistant sans déployer immédiatement un cluster, mais à la place, l'assistant crée un fichier de configuration à partir duquel le cluster peut être déployé ultérieurement. Ceci est très utile si vous devez modifier les valeurs par défaut de l'IPI ou si vous souhaitez déployer plusieurs clusters identiques dans votre environnement pour d'autres utilisations telles que le multi-hébergement. Pour plus d'informations sur la création d'une configuration d'installation personnalisée pour OpenShift, consultez"Red Hat OpenShift Installation d'un cluster sur vSphere avec personnalisations" .

Service Red Hat OpenShift sur AWS

Red Hat OpenShift Service sur AWS (ROSA) est un service géré que vous pouvez utiliser pour créer, mettre à l'échelle et déployer des applications conteneurisées avec la plateforme Kubernetes d'entreprise Red Hat OpenShift sur AWS. ROSA rationalise le déplacement des charges de travail Red Hat OpenShift sur site vers AWS et offre une intégration étroite avec d'autres services AWS.

Pour plus d'informations sur ROSA, consultez la documentation ici :"Service Red Hat OpenShift sur AWS (documentation AWS)" . "Service Red Hat OpenShift sur AWS (documentation Red Hat)" .

Systèmes de stockage NetApp

NetApp ONTAP

NetApp ONTAP est un puissant outil logiciel de stockage doté de fonctionnalités telles

qu’une interface graphique intuitive, des API REST avec intégration d’automatisation, des analyses prédictives et des mesures correctives basées sur l’IA, des mises à niveau matérielles non perturbatrices et une importation inter-stockage.

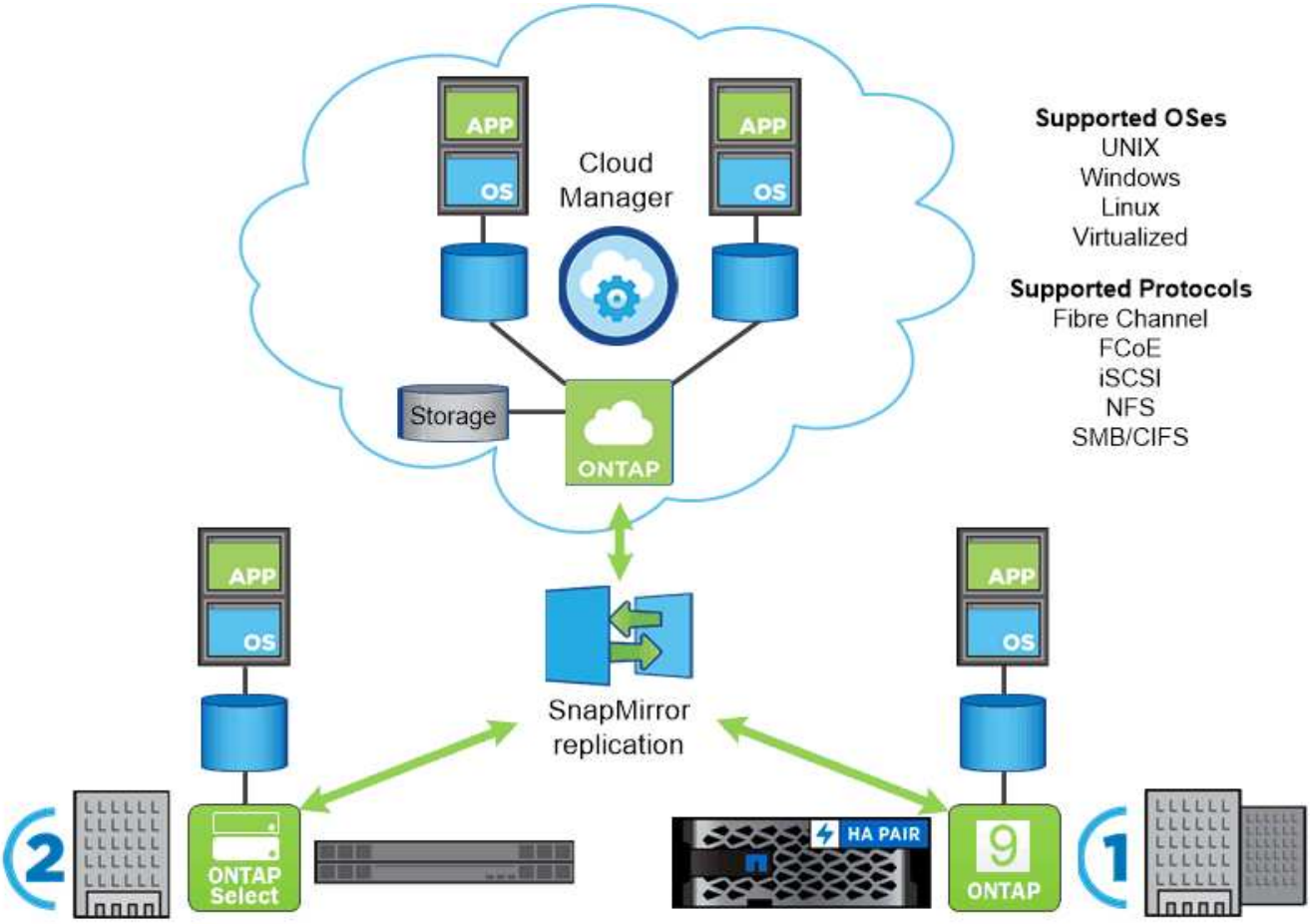
Pour plus d’informations sur le système de stockage NetApp ONTAP , visitez le "Site Web NetApp ONTAP" .

ONTAP offre les fonctionnalités suivantes :

- Un système de stockage unifié avec accès simultané aux données et gestion des protocoles NFS, CIFS, iSCSI, FC, FCoE et FC-NVMe.
- Différents modèles de déploiement incluent des configurations matérielles entièrement flash, hybrides et entièrement HDD sur site ; des plates-formes de stockage basées sur des machines virtuelles sur un hyperviseur pris en charge tel qu’ONTAP ONTAP Select; et dans le cloud sous Cloud Volumes ONTAP.
- Efficacité de stockage des données accrue sur les systèmes ONTAP avec prise en charge de la hiérarchisation automatique des données, de la compression des données en ligne, de la déduplication et du compactage.
- Stockage basé sur la charge de travail et contrôlé par la qualité de service.
- Intégration transparente avec un cloud public pour la hiérarchisation et la protection des données. ONTAP offre également des fonctionnalités robustes de protection des données qui le distinguent dans n’importe quel environnement :
  - * Copies instantanées NetApp .* Une sauvegarde rapide et ponctuelle des données utilisant une quantité minimale d’espace disque sans surcharge de performances supplémentaire.
  - * NetApp SnapMirror.* Met en miroir les copies instantanées des données d’un système de stockage vers un autre. ONTAP prend également en charge la mise en miroir des données sur d’autres plates-formes physiques et services cloud natifs.
  - * NetApp SnapLock.* Gestion efficace des données non réinscriptibles en les écrivant sur des volumes spéciaux qui ne peuvent pas être écrasés ou effacés pendant une période déterminée.
  - * NetApp SnapVault.* Sauvegarde les données de plusieurs systèmes de stockage vers une copie instantanée centrale qui sert de sauvegarde pour tous les systèmes désignés.
  - * NetApp SyncMirror.* Fournit une mise en miroir en temps réel au niveau RAID des données sur deux plex de disques différents connectés physiquement au même contrôleur.
  - * NetApp SnapRestore.* Fournit une restauration rapide des données sauvegardées à la demande à partir de copies instantanées.
  - * NetApp FlexClone.* Fournit le provisionnement instantané d’une copie entièrement lisible et inscriptible d’un volume NetApp basé sur une copie Snapshot.

Pour plus d’informations sur ONTAP, consultez le "Centre de documentation ONTAP 9" .

NetApp ONTAP est disponible sur site, virtualisé ou dans le cloud.



Plateformes NetApp

NetApp AFF/ FAS

NetApp fournit des plates-formes de stockage robustes entièrement flash (AFF) et hybrides évolutives (FAS), conçues sur mesure avec des performances à faible latence, une protection des données intégrée et une prise en charge multiprotocole.

Les deux systèmes sont alimentés par le logiciel de gestion de données NetApp ONTAP , le logiciel de gestion de données le plus avancé du secteur pour une gestion du stockage simplifiée, intégrée au cloud et hautement disponible afin de fournir une vitesse, une efficacité et une sécurité de niveau entreprise dont votre infrastructure de données a besoin.

Pour plus d'informations sur les plateformes NETAPP AFF/ FAS , cliquez sur "ici" .

ONTAP Select

ONTAP Select est un déploiement défini par logiciel de NetApp ONTAP qui peut être déployé sur un hyperviseur dans votre environnement. Il peut être installé sur VMware vSphere ou sur KVM et offre toutes les fonctionnalités et l'expérience d'un système ONTAP basé sur le matériel.

Pour plus d'informations sur ONTAP Select, cliquez sur "ici" .

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP est une version déployée dans le cloud de NetApp ONTAP disponible pour être déployée dans un certain nombre de clouds publics, notamment : Amazon AWS, Microsoft Azure et Google Cloud.

Pour plus d'informations sur Cloud Volumes ONTAP, cliquez sur "ici" .

Amazon FSx ONTAP

Amazon FSx ONTAP fournit un stockage partagé entièrement géré dans le cloud AWS avec les capacités populaires d'accès et de gestion des données d' ONTAP. Pour plus d'informations sur Amazon FSx ONTAP, cliquez sur "ici" .

Azure NetApp Files

Azure NetApp Files est un service de stockage de fichiers Azure natif, propriétaire, de classe entreprise et hautes performances. Il fournit des volumes en tant que service pour lesquels vous pouvez créer des comptes NetApp , des pools de capacité et des volumes. Vous pouvez également sélectionner les niveaux de service et de performance et gérer la protection des données. Vous pouvez créer et gérer des partages de fichiers hautes performances, hautement disponibles et évolutifs en utilisant les mêmes protocoles et outils que ceux que vous connaissez et sur lesquels vous comptez sur site. Pour plus d'informations sur Azure NetApp Files, cliquez sur "ici" .

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes est un service de stockage de données entièrement géré et basé sur le cloud qui offre des capacités de gestion de données avancées et des performances hautement évolutives. Il vous permet de déplacer des applications basées sur des fichiers vers Google Cloud. Il prend en charge les protocoles Network File System (NFSv3 et NFSv4.1) et Server Message Block (SMB) intégrés, vous n'avez donc pas besoin de réarchitecturer vos applications et pouvez continuer à obtenir un stockage persistant pour vos applications. Pour plus d'informations sur Google Cloud NetApp VolumesP, cliquez sur "ici" .

NetApp Element: Red Hat OpenShift avec NetApp

Le logiciel NetApp Element offre des performances modulaires et évolutives, chaque nœud de stockage offrant une capacité et un débit garantis à l'environnement. Les systèmes NetApp Element peuvent évoluer de 4 à 100 nœuds dans un seul cluster et offrent un certain nombre de fonctionnalités avancées de gestion du stockage.

Pour plus d'informations sur les systèmes de stockage NetApp Element , visitez le "Site Web de NetApp Solidfire" .

Redirection de connexion iSCSI et capacités d'auto-réparation

Le logiciel NetApp Element exploite le protocole de stockage iSCSI, une méthode standard pour encapsuler les commandes SCSI sur un réseau TCP/IP traditionnel. Lorsque les normes SCSI changent ou lorsque les performances des réseaux Ethernet s'améliorent, le protocole de stockage iSCSI en bénéficie sans qu'aucune modification ne soit nécessaire.

Bien que tous les nœuds de stockage disposent d'une adresse IP de gestion et d'une adresse IP de stockage, le logiciel NetApp Element annonce une seule adresse IP virtuelle de stockage (adresse SVIP) pour tout le trafic de stockage du cluster. Dans le cadre du processus de connexion iSCSI, le stockage peut répondre que le volume cible a été déplacé vers une adresse différente et qu'il ne peut donc pas poursuivre le processus de négociation. L'hôte réémet ensuite la demande de connexion à la nouvelle adresse dans un processus qui ne nécessite aucune reconfiguration côté hôte. Ce processus est connu sous le nom de redirection de connexion iSCSI.

La redirection de connexion iSCSI est un élément clé du cluster logiciel NetApp Element . Lorsqu'une demande de connexion à l'hôte est reçue, le nœud décide quel membre du cluster doit gérer le trafic en fonction des IOPS et des exigences de capacité du volume. Les volumes sont répartis sur le cluster logiciel NetApp Element et sont redistribués si un seul nœud gère trop de trafic pour ses volumes ou si un nouveau nœud est ajouté. Plusieurs copies d'un volume donné sont allouées sur la matrice.

De cette manière, si une panne de nœud est suivie d'une redistribution de volume, il n'y a aucun effet sur la connectivité de l'hôte au-delà d'une déconnexion et d'une connexion avec redirection vers le nouvel emplacement. Avec la redirection de connexion iSCSI, un cluster logiciel NetApp Element est une architecture évolutive et auto-réparatrice capable de réaliser des mises à niveau et des opérations sans interruption.

QoS du cluster logiciel NetApp Element

Un cluster logiciel NetApp Element permet de configurer dynamiquement la qualité de service en fonction du volume. Vous pouvez utiliser les paramètres QoS par volume pour contrôler les performances de stockage en fonction des SLA que vous définissez. Les trois paramètres configurables suivants définissent la qualité de service :

- **IOPS minimum.** Nombre minimal d'IOPS soutenus que le cluster logiciel NetApp Element fournit à un volume. Le nombre minimum d'IOPS configuré pour un volume est le niveau de performance garanti pour un volume. Les performances par volume ne descendent pas en dessous de ce niveau.
- **IOPS maximum.** Nombre maximal d'IOPS soutenus que le cluster logiciel NetApp Element fournit à un volume particulier.
- **IOPS en rafale.** Le nombre maximal d'IOPS autorisé dans un scénario de rafale courte. Le paramètre de durée de rafale est configurable, avec une valeur par défaut de 1 minute. Si un volume a fonctionné en dessous du niveau d'IOPS maximal, des crédits d'éclatement sont accumulés. Lorsque les niveaux de performances deviennent très élevés et sont poussés, de courtes rafales d'IOPS au-delà des IOPS maximales sont autorisées sur le volume.

Multilocation

La multilocation sécurisée est obtenue grâce aux fonctionnalités suivantes :

- **Authentification sécurisée.** Le protocole d'authentification Challenge-Handshake (CHAP) est utilisé pour l'accès sécurisé aux volumes. Le protocole LDAP (Lightweight Directory Access Protocol) est utilisé pour un accès sécurisé au cluster à des fins de gestion et de reporting.
- **Groupes d'accès aux volumes (VAG).** En option, les VAG peuvent être utilisés à la place de l'authentification, en mappant n'importe quel nombre de noms qualifiés iSCSI (IQN) spécifiques à l'initiateur iSCSI à un ou plusieurs volumes. Pour accéder à un volume dans un VAG, l'IQN de l'initiateur

doit figurer dans la liste des IQN autorisés pour le groupe de volumes.

- **Réseaux locaux virtuels locataires (VLAN).** Au niveau du réseau, la sécurité du réseau de bout en bout entre les initiateurs iSCSI et le cluster logiciel NetApp Element est facilitée par l'utilisation de VLAN. Pour tout VLAN créé pour isoler une charge de travail ou un locataire, NetApp Element Software crée une adresse SVIP cible iSCSI distincte qui est accessible uniquement via le VLAN spécifique.
- **VLAN compatibles VRF.** Pour renforcer la sécurité et l'évolutivité du centre de données, le logiciel NetApp Element vous permet d'activer n'importe quel VLAN locataire pour une fonctionnalité de type VRF. Cette fonctionnalité ajoute ces deux capacités clés :
  - **Routage L3 vers une adresse SVIP de locataire.** Cette fonctionnalité vous permet de situer les initiateurs iSCSI sur un réseau ou un VLAN distinct de celui du cluster logiciel NetApp Element .
  - **Sous-réseaux IP superposés ou en double.** Cette fonctionnalité vous permet d'ajouter un modèle aux environnements locataires, permettant à chaque VLAN locataire respectif de se voir attribuer des adresses IP à partir du même sous-réseau IP. Cette capacité peut être utile pour les environnements de fournisseurs de services dans lesquels l'échelle et la préservation de l'espace IP sont importantes.

Efficacité du stockage d'entreprise

Le cluster logiciel NetApp Element augmente l'efficacité et les performances globales du stockage. Les fonctionnalités suivantes sont exécutées en ligne, sont toujours activées et ne nécessitent aucune configuration manuelle par l'utilisateur :

- **Déduplication.** Le système ne stocke que des blocs 4K uniques. Tous les blocs 4K en double sont automatiquement associés à une version déjà stockée des données. Les données sont stockées sur des lecteurs de blocs et sont mises en miroir à l'aide du logiciel de protection des données Helix de NetApp Element . Ce système réduit considérablement la consommation de capacité et les opérations d'écriture au sein du système.
- **Compression.** La compression est effectuée en ligne avant que les données ne soient écrites dans la NVRAM. Les données sont compressées, stockées dans des blocs de 4 Ko et restent compressées dans le système. Cette compression réduit considérablement la consommation de capacité, les opérations d'écriture et la consommation de bande passante sur l'ensemble du cluster.
- **Provisionnement léger.** Cette capacité fournit la quantité de stockage appropriée au moment où vous en avez besoin, éliminant ainsi la consommation de capacité causée par des volumes surprovisionnés ou sous-utilisés.
- **Hélix.** Les métadonnées d'un volume individuel sont stockées sur un lecteur de métadonnées et sont répliquées sur un lecteur de métadonnées secondaire à des fins de redondance.

> Element a été conçu pour l'automatisation. Toutes les fonctionnalités de stockage sont disponibles via les API. Ces API sont la seule méthode utilisée par l'interface utilisateur pour contrôler le système.

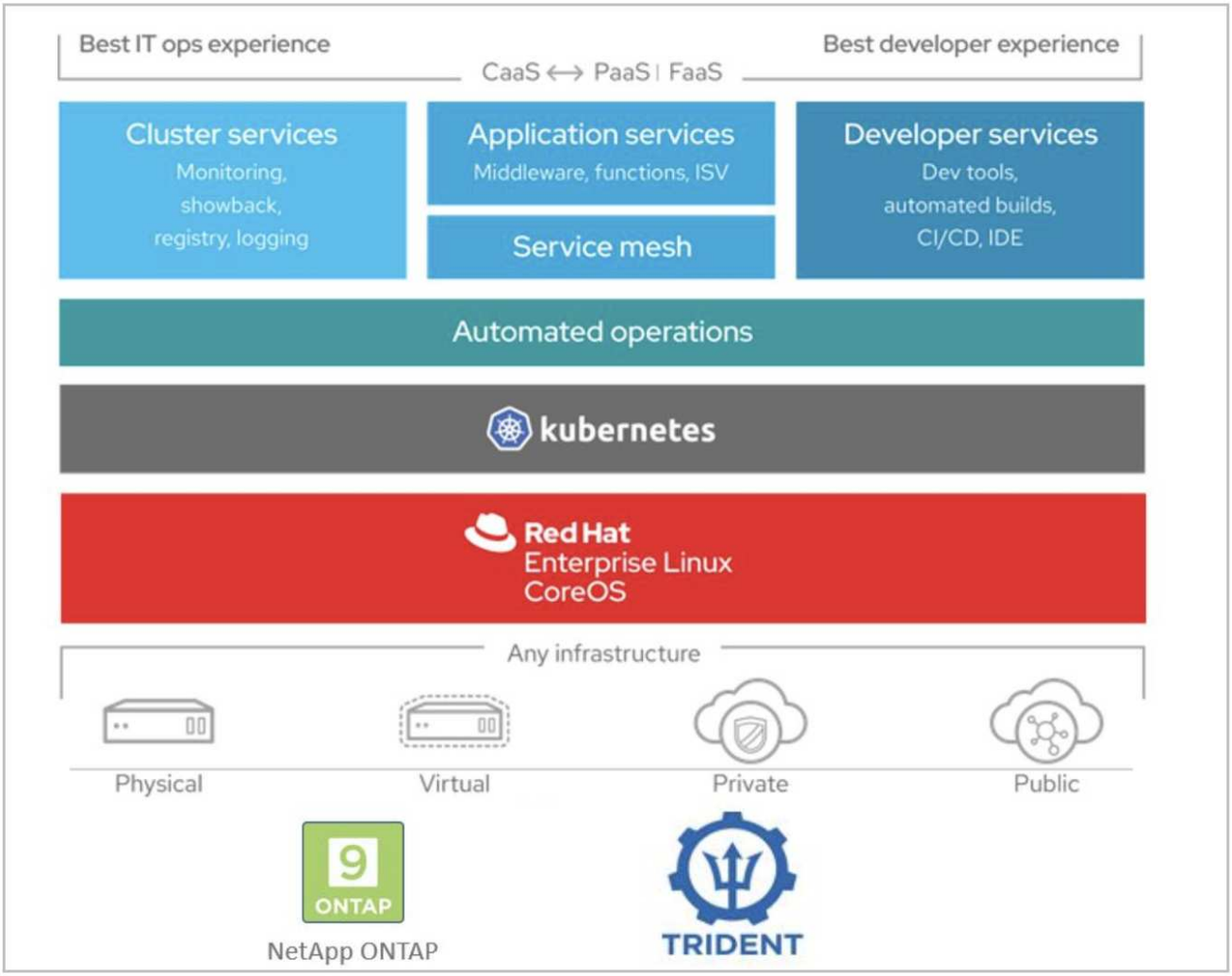
Intégrations de stockage NetApp

En savoir plus sur l'intégration de NetApp Trident avec Red Hat OpenShift

Découvrez NetApp Trident Protect, qui a été validé pour la gestion des applications et du stockage persistant pour la solution de virtualisation OpenShift.

Trident, un provisionneur et orchestrateur de stockage open source maintenu par NetApp et NetApp Trident Protect, vous aide à orchestrer et à gérer les données persistantes dans des environnements basés sur des

conteneurs, tels que Red Hat OpenShift.



Les pages suivantes contiennent des informations supplémentaires sur les produits NetApp qui ont été validés pour la gestion des applications et du stockage persistant dans la solution Red Hat OpenShift avec NetApp :

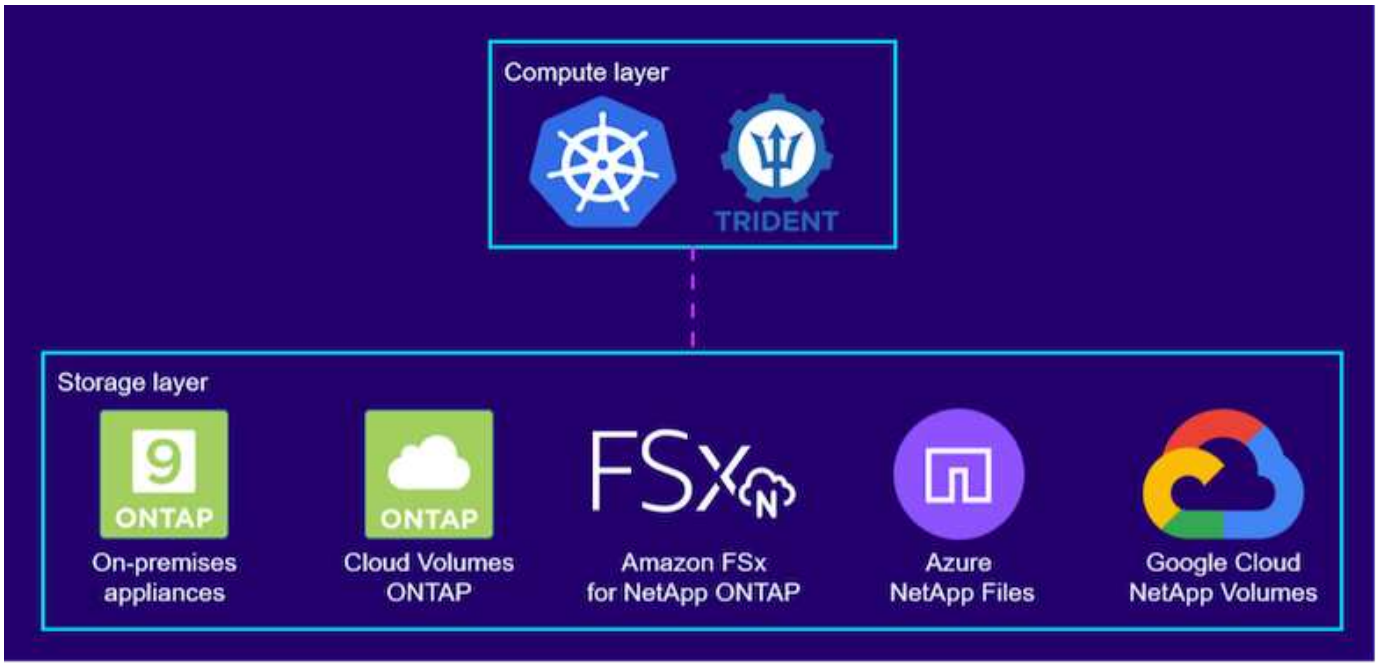
- "Documentation Trident"
- "Documentation de protection Trident"

NetApp Trident

Présentation de Trident

Trident est un orchestrateur de stockage open source et entièrement pris en charge pour les conteneurs et les distributions Kubernetes, y compris Red Hat OpenShift. Trident fonctionne avec l'ensemble du portefeuille de stockage NetApp , y compris les systèmes de stockage NetApp ONTAP et Element, et prend également en charge les connexions NFS et iSCSI. Trident accélère le flux de travail DevOps en permettant aux utilisateurs finaux de provisionner et de gérer le stockage à partir de leurs systèmes de stockage NetApp sans nécessiter l'intervention d'un administrateur de stockage.

Un administrateur peut configurer un certain nombre de backends de stockage en fonction des besoins du projet et des modèles de système de stockage qui permettent des fonctionnalités de stockage avancées, notamment la compression, des types de disques spécifiques ou des niveaux de qualité de service qui garantissent un certain niveau de performances. Une fois définis, ces backends peuvent être utilisés par les développeurs dans leurs projets pour créer des revendications de volume persistant (PVC) et pour attacher un stockage persistant à leurs conteneurs à la demande.



Trident a un cycle de développement rapide et, tout comme Kubernetes, est publié quatre fois par an.

Une matrice de support indiquant quelle version de Trident a été testée avec quelle distribution Kubernetes peut être trouvée "ici" .

Veuillez vous référer à la"Documentation du produit Trident" pour les détails d'installation et de configuration.

Télécharger Trident

Pour installer Trident sur le cluster utilisateur déployé et provisionner un volume persistant, procédez comme suit :

1. Téléchargez l'archive d'installation sur le poste de travail administrateur et extrayez le contenu. La version actuelle de Trident peut être téléchargée "ici" .
2. Extrayez l'installation de Trident à partir du bundle téléchargé.

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

Installer l'opérateur Trident avec Helm

1. Définissez d'abord l'emplacement du cluster d'utilisateurs `kubeconfig` fichier comme variable d'environnement afin que vous n'ayez pas à y faire référence, car Trident n'a pas la possibilité de transmettre ce fichier.

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

2. Exécutez la commande Helm pour installer l'opérateur Trident à partir de l'archive tar dans le répertoire helm tout en créant l'espace de noms trident dans votre cluster utilisateur.

```
[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. Vous pouvez vérifier que Trident est correctement installé en vérifiant les pods qui s'exécutent dans l'espace de noms ou en utilisant le binaire tridentctl pour vérifier la version installée.

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

Dans certains cas, les environnements clients peuvent nécessiter la personnalisation du déploiement Trident . Dans ces cas, il est également possible d'installer manuellement l'opérateur Trident et de mettre à jour les manifestes inclus pour personnaliser le déploiement.

Installer manuellement l'opérateur Trident

1. Tout d'abord, définissez l'emplacement du cluster d'utilisateurs `kubeconfig` fichier comme variable d'environnement afin que vous n'ayez pas à y faire référence, car Trident n'a pas la possibilité de transmettre ce fichier.

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

2. Le `trident-installer` le répertoire contient des manifestes pour définir toutes les ressources requises. En utilisant les manifestes appropriés, créez le `TridentOrchestrator` définition de ressource personnalisée.

```
[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. S'il n'en existe pas, créez un espace de noms Trident dans votre cluster à l'aide du manifeste fourni.

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

4. Créez les ressources nécessaires au déploiement de l'opérateur Trident , comme un `ServiceAccount`

pour l'opérateur, un `ClusterRole` et `ClusterRoleBinding` au `ServiceAccount` , un dédié `PodSecurityPolicy` , ou l'opérateur lui-même.

```
[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. Vous pouvez vérifier l'état de l'opérateur après son déploiement avec les commandes suivantes :

```
[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. Avec l'opérateur déployé, nous pouvons maintenant l'utiliser pour installer Trident. Cela nécessite de créer un `TridentOrchestrator` .

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

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

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

7. Vous pouvez vérifier que Trident est correctement installé en vérifiant les pods qui s'exécutent dans l'espace de noms ou en utilisant le binaire tridentctl pour vérifier la version installée.

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

Préparer les nœuds de travail pour le stockage

NFS

La plupart des distributions Kubernetes sont livrées avec les packages et utilitaires permettant de monter les backends NFS installés par défaut, y compris Red Hat OpenShift.

Cependant, pour NFSv3, il n'existe aucun mécanisme permettant de négocier la concurrence entre le client et le serveur. Par conséquent, le nombre maximal d'entrées de la table d'emplacements sunrpc côté client doit

être synchronisé manuellement avec la valeur prise en charge sur le serveur pour garantir les meilleures performances pour la connexion NFS sans que le serveur n'ait à réduire la taille de la fenêtre de connexion.

Pour ONTAP, le nombre maximal d'entrées de table d'emplacements sunrpc prises en charge est de 128, c'est-à-dire ONTAP peut traiter 128 requêtes NFS simultanées à la fois. Cependant, par défaut, Red Hat CoreOS/Red Hat Enterprise Linux dispose d'un maximum de 65 536 entrées de table d'emplacements sunrpc par connexion. Nous devons définir cette valeur sur 128 et cela peut être fait à l'aide de Machine Config Operator (MCO) dans OpenShift.

Pour modifier le nombre maximal d'entrées de la table d'emplacements sunrpc dans les nœuds de travail OpenShift, procédez comme suit :

1. Connectez-vous à la console Web OCP et accédez à Calcul > Configurations de la machine. Cliquez sur Créer une configuration de machine. Copiez et collez le fichier YAML et cliquez sur Créer.

```
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. Une fois le MCO créé, la configuration doit être appliquée sur tous les nœuds de travail et redémarrée un par un. L'ensemble du processus prend environ 20 à 30 minutes. Vérifiez si la configuration de la machine est appliquée en utilisant `oc get mcp` et assurez-vous que le pool de configuration de la machine pour les travailleurs est mis à jour.

```
[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

Pour préparer les nœuds de travail afin de permettre le mappage des volumes de stockage de blocs via le protocole iSCSI, vous devez installer les packages nécessaires pour prendre en charge cette fonctionnalité.

Dans Red Hat OpenShift, cela est géré en appliquant un MCO (Machine Config Operator) à votre cluster après son déploiement.

Pour configurer les nœuds de travail afin d'exécuter les services iSCSI, procédez comme suit :

1. Connectez-vous à la console Web OCP et accédez à Calcul > Configurations de la machine. Cliquez sur Créer une configuration de machine. Copiez et collez le fichier YAML et cliquez sur Créer.

   Lorsque vous n'utilisez pas le multipathing :

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

   Lors de l'utilisation du 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. Une fois la configuration créée, il faut environ 20 à 30 minutes pour appliquer la configuration aux nœuds de travail et les recharger. Vérifiez si la configuration de la machine est appliquée en utilisant `oc get mcp` et assurez-vous que le pool de configuration de la machine pour les travailleurs est mis à jour. Vous pouvez également vous connecter aux nœuds de travail pour confirmer que le service iscsid est en cours d'exécution (et que le service multipathd est en cours d'exécution si vous utilisez le multipathing).

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

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

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

Il est également possible de confirmer que MachineConfig a été appliqué avec succès et que les services ont été démarrés comme prévu en exécutant le `oc debug` commande avec les drapeaux appropriés.

Créer des backends de système de stockage

Une fois l'installation de Trident Operator terminée, vous devez configurer le backend pour la plate-forme de

stockage NetApp spécifique que vous utilisez. Suivez les liens ci-dessous afin de continuer l'installation et la configuration de Trident.

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

Configuration NFS de NetApp ONTAP

Pour activer l'intégration de Trident avec le système de stockage NetApp ONTAP , vous devez créer un backend qui permet la communication avec le système de stockage.

1. Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp ONTAP servant NFS, copiez le `backend-ontap-nas.json` fichier dans votre répertoire de travail et modifiez le fichier.

```
[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. Modifiez les valeurs backendName, managementLIF, dataLIF, svm, username et password dans ce fichier.

```
{
    "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"
}
```

> Il est recommandé de définir la valeur backendName personnalisée comme une combinaison du storageDriverName et du dataLIF qui sert NFS pour une identification facile.

3. Avec ce fichier backend en place, exécutez la commande suivante pour créer votre premier backend.

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

4. Une fois le backend créé, vous devez ensuite créer une classe de stockage. Tout comme pour le backend, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier sample-inputs. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le backend créé.

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

5. La seule modification qui doit être apportée à ce fichier est de définir le `backendType` valeur au nom du pilote de stockage du backend nouvellement créé. Notez également la valeur du champ de nom, qui doit être référencée dans une étape ultérieure.

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

> Il existe un champ facultatif appelé `fsType` qui est défini dans ce fichier. Cette ligne peut être supprimée dans les backends NFS.

6. Exécutez le `oc` commande pour créer la classe de stockage.

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

7. Une fois la classe de stockage créée, vous devez ensuite créer la première revendication de volume persistant (PVC). Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action situé également dans sample-inputs.

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

8. La seule modification qui doit être apportée à ce fichier est de s'assurer que le `storageClassName` le champ correspond à celui qui vient d'être créé. La définition du PVC peut être davantage personnalisée selon les besoins de la charge de travail à provisionner.

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

9. Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de support en cours de création, vous pouvez donc observer le processus au fur et à mesure de son exécution.

```
[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
```

Configuration iSCSI de NetApp ONTAP

Pour activer l'intégration de Trident avec le système de stockage NetApp ONTAP , vous devez créer un backend qui permet la communication avec le système de stockage.

1. Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp ONTAP servant iSCSI, copiez le

`backend-ontap-san.json` fichier dans votre répertoire de travail et modifiez le fichier.

```
[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. Modifiez les valeurs managementLIF, dataLIF, svm, username et password dans ce fichier.

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

3. Avec ce fichier backend en place, exécutez la commande suivante pour créer votre premier backend.

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

4. Une fois le backend créé, vous devez ensuite créer une classe de stockage. Tout comme pour le backend, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier sample-inputs. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le backend créé.

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

5. La seule modification qui doit être apportée à ce fichier est de définir le `backendType` valeur au nom du

pilote de stockage du backend nouvellement créé. Notez également la valeur du champ de nom, qui doit être référencée dans une étape ultérieure.

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

Il existe un champ facultatif appelé `fsType` qui est défini dans ce fichier. Dans les backends iSCSI, cette valeur peut être définie sur un type de système de fichiers Linux spécifique (XFS, ext4, etc.) ou peut être supprimée pour permettre à OpenShift de décider quel système de fichiers utiliser.

6. Exécutez le `oc` commande pour créer la classe de stockage.

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

7. Une fois la classe de stockage créée, vous devez ensuite créer la première revendication de volume persistant (PVC). Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action situé également dans sample-inputs.

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

8. La seule modification qui doit être apportée à ce fichier est de s'assurer que le `storageClassName` le champ correspond à celui qui vient d'être créé. La définition du PVC peut être davantage personnalisée selon les besoins de la charge de travail à provisionner.

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

9. Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de support en cours de création, vous pouvez donc observer le processus au fur et à mesure de son exécution.

```
[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
```

Configuration iSCSI de NetApp Element

Pour activer l'intégration de Trident avec le système de stockage NetApp Element , vous devez créer un backend qui permet la communication avec le système de stockage à l'aide du protocole iSCSI.

1. Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp Element servant iSCSI, copiez le `backend-solidfire.json` fichier dans votre répertoire de travail et modifiez le fichier.

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

   a. Modifiez l'utilisateur, le mot de passe et la valeur MVIP sur le `EndPoint` doubler.

   b. Modifier le `SVIP` valeur.

```
 {
    "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. Avec ce fichier back-end en place, exécutez la commande suivante pour créer votre premier back-end.

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

3. Une fois le backend créé, vous devez ensuite créer une classe de stockage. Tout comme pour le backend, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier sample-inputs. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le backend créé.

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

4. La seule modification qui doit être apportée à ce fichier est de définir le `backendType` valeur au nom du pilote de stockage du backend nouvellement créé. Notez également la valeur du champ de nom, qui doit être référencée dans une étape ultérieure.

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

> Il existe un champ facultatif appelé `fsType` qui est défini dans ce fichier. Dans les backends iSCSI, cette valeur peut être définie sur un type de système de fichiers Linux spécifique (XFS, ext4, etc.), ou elle peut être supprimée pour permettre à OpenShift de décider quel système de fichiers utiliser.

5. Exécutez le `oc` commande pour créer la classe de stockage.

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

6. Une fois la classe de stockage créée, vous devez ensuite créer la première revendication de volume persistant (PVC). Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action situé également dans sample-inputs.

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

7. La seule modification qui doit être apportée à ce fichier est de s'assurer que le `storageClassName` le champ correspond à celui qui vient d'être créé. La définition du PVC peut être davantage personnalisée selon les besoins de la charge de travail à provisionner.

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

8. Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de support en cours de création, vous pouvez donc observer le processus au fur et à mesure de son exécution.

```
[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
```

Options de configuration avancées

Explorer les options d'équilibrage de charge

Explorer les options d'équilibrage de charge : Red Hat OpenShift avec NetApp

Dans la plupart des cas, Red Hat OpenShift rend les applications accessibles au monde extérieur via des itinéraires. Un service est exposé en lui donnant un nom d'hôte accessible de l'extérieur. L'itinéraire défini et les points de terminaison identifiés par son service peuvent être consommés par un routeur OpenShift pour fournir cette connectivité nommée aux clients externes.

Cependant, dans certains cas, les applications nécessitent le déploiement et la configuration d'équilibreurs de charge personnalisés pour exposer les services appropriés. NetApp Trident Protect en est un exemple. Pour répondre à ce besoin, nous avons évalué un certain nombre d'options d'équilibrage de charge personnalisées. Leur installation et leur configuration sont décrites dans cette section.

Les pages suivantes contiennent des informations supplémentaires sur les options d'équilibrage de charge validées dans la solution Red Hat OpenShift avec NetApp :

- "MetalLB"
- "F5 BIG-IP"

Installation des équilibreurs de charge MetalLB : Red Hat OpenShift avec NetApp

Cette page répertorie les instructions d'installation et de configuration de l'équilibreur de charge MetalLB.

MetalLB est un équilibreur de charge réseau auto-hébergé installé sur votre cluster OpenShift qui permet la création de services OpenShift de type équilibreur de charge dans des clusters qui ne s'exécutent pas sur un fournisseur de cloud. Les deux principales fonctionnalités de MetalLB qui fonctionnent ensemble pour prendre en charge les services LoadBalancer sont l'allocation d'adresses et l'annonce externe.

Options de configuration de MetalLB

En fonction de la manière dont MetalLB annonce l'adresse IP attribuée aux services LoadBalancer en dehors du cluster OpenShift, il fonctionne selon deux modes :

- **Mode couche 2.** Dans ce mode, un nœud du cluster OpenShift prend possession du service et répond aux requêtes ARP pour cette IP afin de la rendre accessible en dehors du cluster OpenShift. Étant donné que seul le nœud annonce l'adresse IP, il présente un goulot d'étranglement de bande passante et des limitations de basculement lent. Pour plus d'informations, consultez la documentation"ici" .
- **Mode BGP.** Dans ce mode, tous les nœuds du cluster OpenShift établissent des sessions de peering BGP avec un routeur et annoncent les itinéraires pour transférer le trafic vers les adresses IP de service. La condition préalable est d'intégrer MetalLB à un routeur de ce réseau. En raison du mécanisme de hachage dans BGP, il présente certaines limitations lorsque le mappage IP vers nœud pour un service change. Pour plus d'informations, reportez-vous à la documentation"ici" .

Pour les besoins de ce document, nous configurons MetalLB en mode couche 2.

Installation de l'équilibreur de charge MetalLB

1. Téléchargez les ressources MetalLB.

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

2. Modifier le fichier `metallb.yaml` et supprimer `spec.template.spec.securityContext` à partir du contrôleur de déploiement et du DaemonSet du haut-parleur.

   Lignes à supprimer :

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

3. Créer le `metallb-system` espace de noms.

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

4. Créez le CR MetalLB.

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

5. Avant de configurer le haut-parleur MetalLB, accordez au haut-parleur DaemonSet des privilèges élevés afin qu'il puisse effectuer la configuration réseau requise pour faire fonctionner les équilibreurs de charge.

```
[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. Configurez MetalLB en créant un `ConfigMap` dans le `metallb-system` espace de noms.

```
[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. Désormais, lorsque les services d'équilibrage de charge sont créés, MetalLB attribue une adresse IP externe aux services et annonce l'adresse IP en répondant aux requêtes ARP.

> Si vous souhaitez configurer MetalLB en mode BGP, ignorez l'étape 6 ci-dessus et suivez la procédure décrite dans la documentation MetalLB"ici" .

Installation des équilibreurs de charge F5 BIG-IP

F5 BIG-IP est un contrôleur de distribution d'applications (ADC) qui offre un large ensemble de services avancés de gestion du trafic et de sécurité de niveau production tels que l'équilibrage de charge L4-L7, le déchargement SSL/TLS, le DNS, le pare-feu et bien d'autres. Ces services augmentent considérablement la disponibilité, la sécurité et les performances de vos applications.

F5 BIG-IP peut être déployé et consommé de différentes manières, sur du matériel dédié, dans le cloud ou en tant qu'appliance virtuelle sur site. Reportez-vous à la documentation ici pour explorer et déployer F5 BIG-IP selon les besoins.

Pour une intégration efficace des services F5 BIG-IP avec Red Hat OpenShift, F5 propose le service BIG-IP Container Ingress (CIS). CIS est installé en tant que pod contrôleur qui surveille l'API OpenShift pour certaines définitions de ressources personnalisées (CRD) et gère la configuration du système F5 BIG-IP. F5 BIG-IP CIS peut être configuré pour contrôler les types de services LoadBalancers et Routes dans OpenShift.

De plus, pour l'allocation automatique d'adresses IP afin de servir le type LoadBalancer, vous pouvez utiliser le contrôleur IPAM F5. Le contrôleur F5 IPAM est installé en tant que pod de contrôleur qui surveille les services OpenShift API pour LoadBalancer avec une annotation ipamLabel pour allouer l'adresse IP à partir d'un pool préconfiguré.

Cette page répertorie les instructions d'installation et de configuration du contrôleur F5 BIG-IP CIS et IPAM. Comme condition préalable, vous devez disposer d'un système F5 BIG-IP déployé et sous licence. Il doit

également être sous licence pour les services SDN, qui sont inclus par défaut avec la licence de base BIG-IP VE.

F5 BIG-IP peut être déployé en mode autonome ou en cluster. Aux fins de cette validation, F5 BIG-IP a été déployé en mode autonome, mais, à des fins de production, il est préférable d'avoir un cluster de BIG-IP pour éviter un point de défaillance unique.

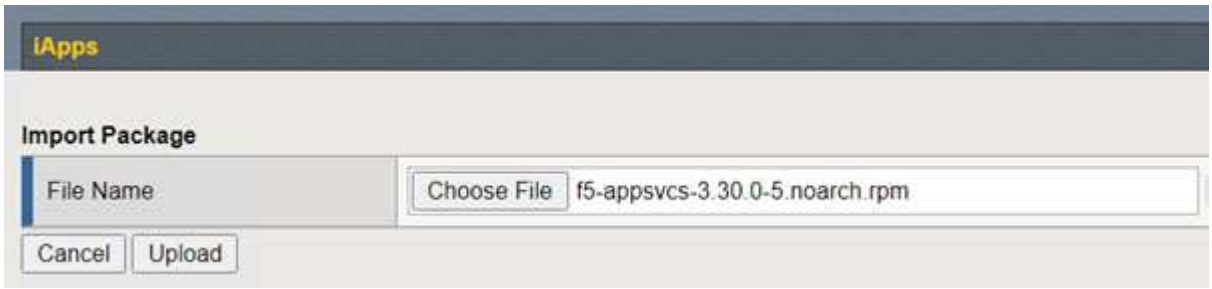
Un système F5 BIG-IP peut être déployé sur du matériel dédié, dans le cloud ou en tant qu'appliance virtuelle sur site avec des versions supérieures à 12.x pour être intégré à F5 CIS. Aux fins du présent document, le système F5 BIG-IP a été validé en tant qu'appareil virtuel, par exemple à l'aide de l'édition BIG-IP VE.

Versions validées

Technologie	Version du logiciel
Red Hat OpenShift	4,6 EUS, 4,7
Édition F5 BIG-IP VE	16.1.0
Service d'entrée de conteneurs F5	2.5.1
Contrôleur IPAM F5	0.1.4
F5 AS3	3.30.0

Installation

1. Installez l'extension F5 Application Services 3 pour permettre aux systèmes BIG-IP d'accepter des configurations au format JSON au lieu de commandes impératives. Aller à "Dépôt GitHub F5 AS3" , et téléchargez le dernier fichier RPM.
2. Connectez-vous au système F5 BIG-IP, accédez à iApps > Package Management LX et cliquez sur Importer.
3. Cliquez sur Choisir un fichier et sélectionnez le fichier AS3 RPM téléchargé, cliquez sur OK, puis sur Télécharger.

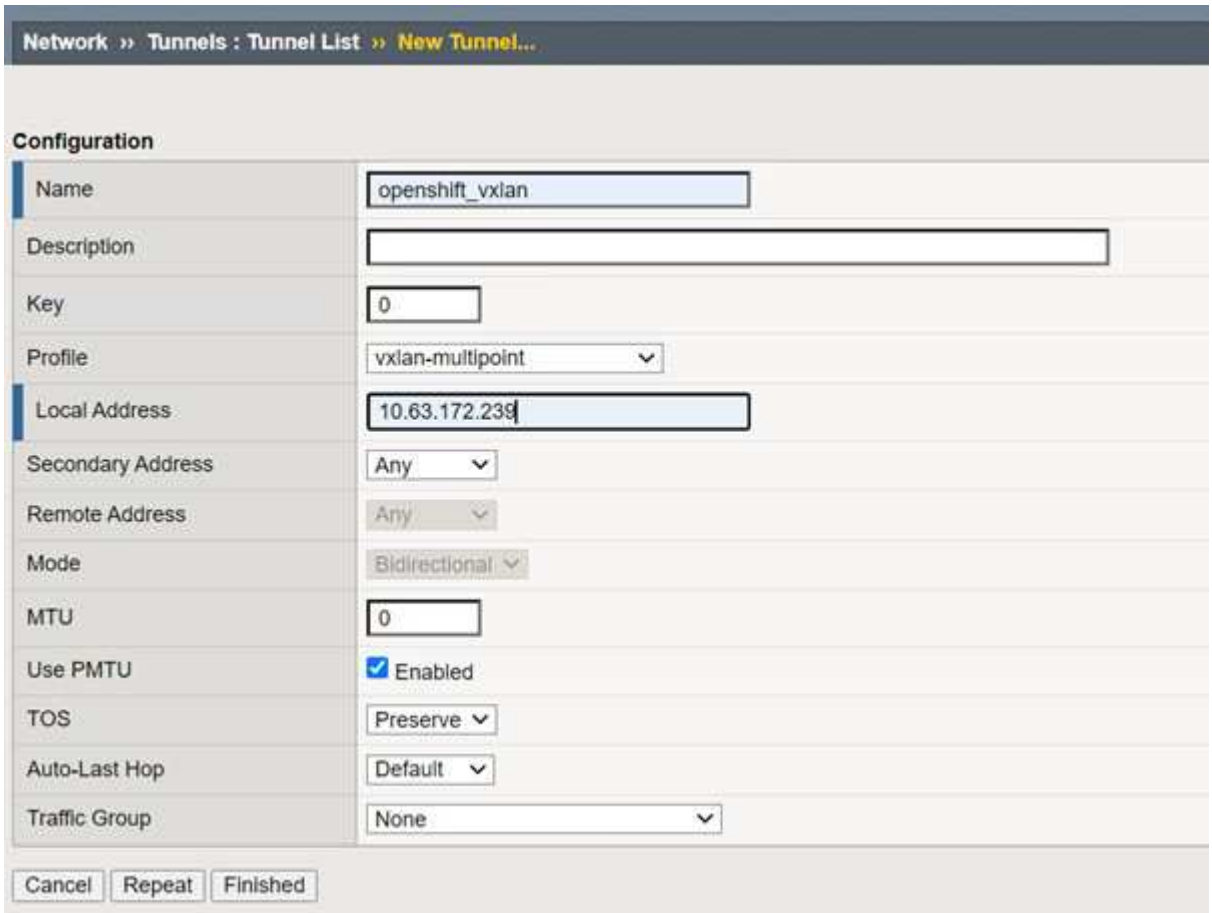


4. Confirmez que l'extension AS3 est installée avec succès.

5. Configurez ensuite les ressources nécessaires à la communication entre les systèmes OpenShift et BIG-

IP. Créez d'abord un tunnel entre OpenShift et le serveur BIG-IP en créant une interface de tunnel VXLAN sur le système BIG-IP pour OpenShift SDN. Accédez à Réseau > Tunnels > Profils, cliquez sur Créer et définissez le profil parent sur vxlan et le type d'inondation sur Multidiffusion. Saisissez un nom pour le profil et cliquez sur Terminé.

6. Accédez à Réseau > Tunnels > Liste des tunnels, cliquez sur Créer et saisissez le nom et l'adresse IP locale du tunnel. Sélectionnez le profil de tunnel créé à l'étape précédente et cliquez sur Terminé.



7. Connectez-vous au cluster Red Hat OpenShift avec les privilèges d'administrateur de cluster.
8. Créez un sous-réseau hôte sur OpenShift pour le serveur F5 BIG-IP, qui étend le sous-réseau du cluster OpenShift au serveur F5 BIG-IP. Téléchargez la définition YAML du sous-réseau hôte.

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

9. Modifiez le fichier de sous-réseau hôte et ajoutez l'adresse IP BIG-IP VTEP (tunnel VXLAN) pour OpenShift SDN.

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

Modifiez l'adresse IP de l'hôte et d'autres détails en fonction de votre environnement.

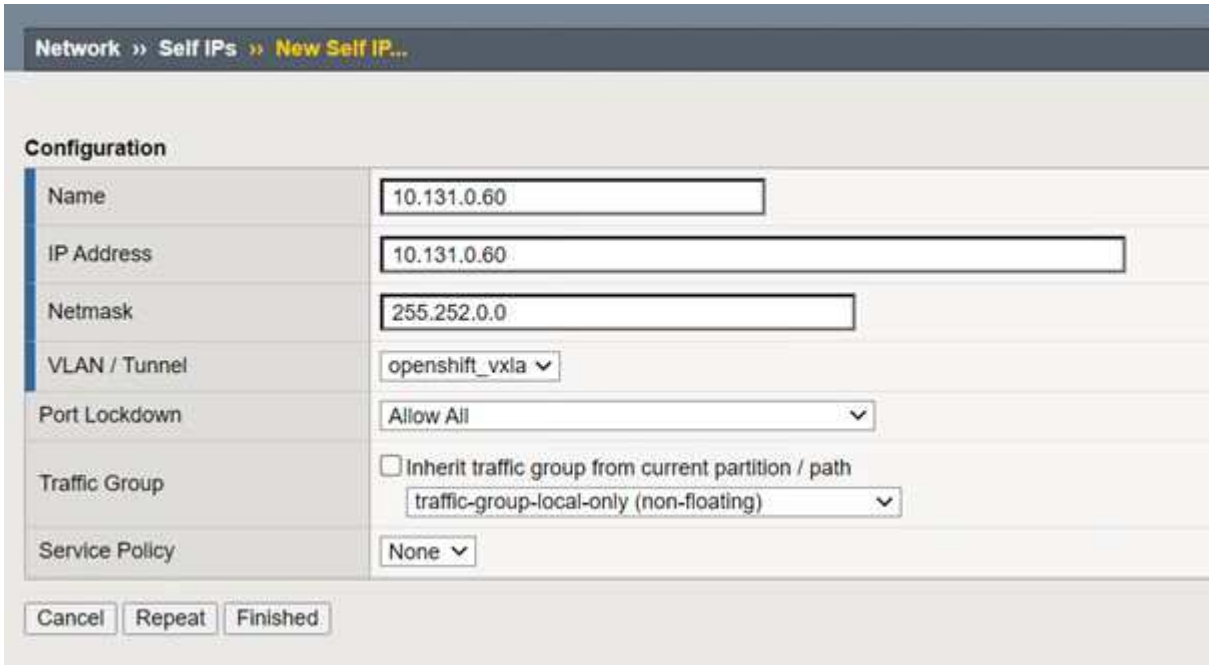
10. Créez la ressource HostSubnet.

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

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

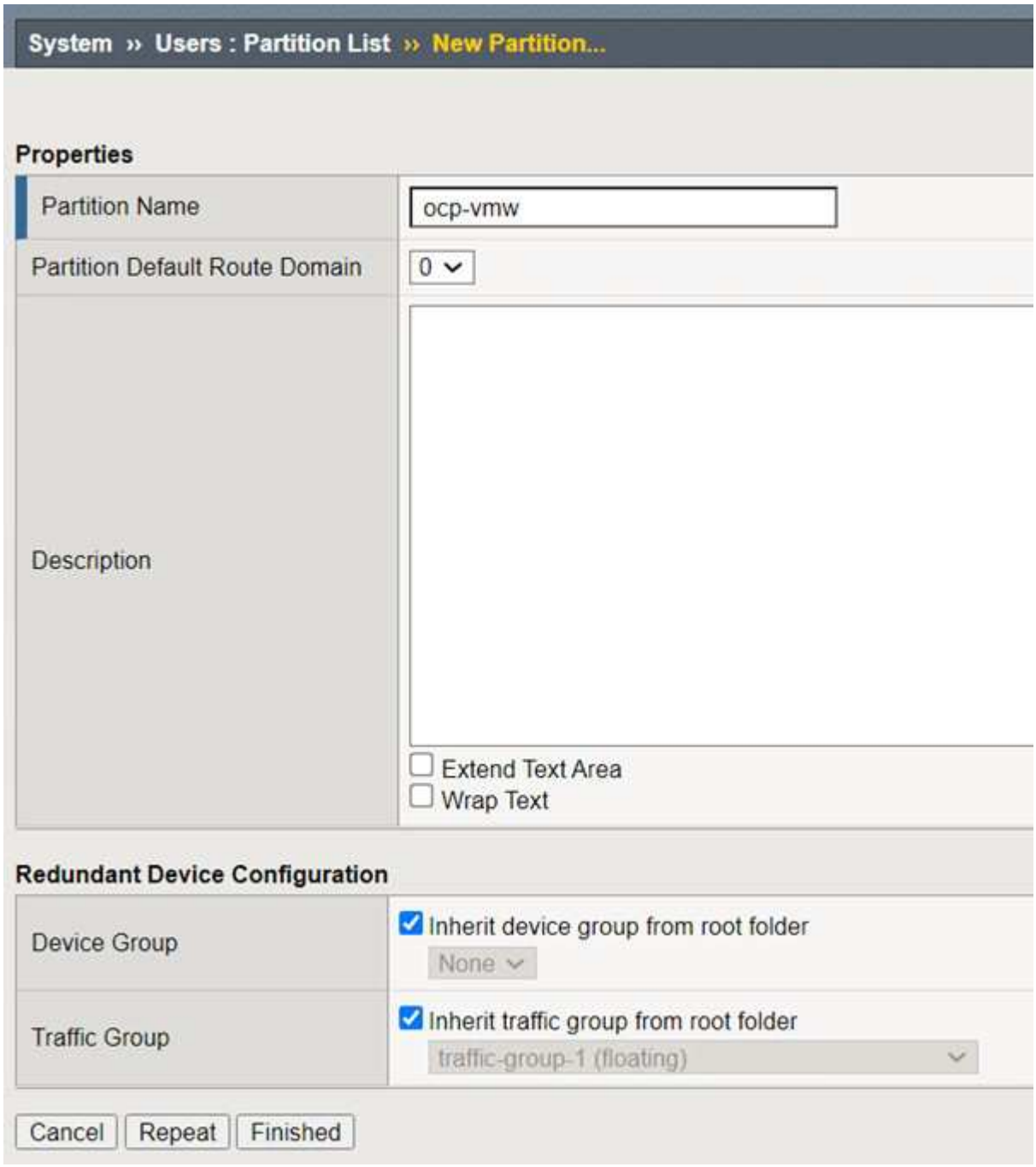
11. Obtenez la plage de sous-réseaux IP du cluster pour le sous-réseau hôte créé pour le serveur F5 BIG-IP.

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

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

12. Créez une IP personnelle sur OpenShift VXLAN avec une IP dans la plage de sous-réseau hôte d'OpenShift correspondant au serveur F5 BIG-IP. Connectez-vous au système F5 BIG-IP, accédez à Réseau > Auto-IP et cliquez sur Créer. Saisissez une adresse IP à partir du sous-réseau IP du cluster créé pour le sous-réseau hôte F5 BIG-IP, sélectionnez le tunnel VXLAN et saisissez les autres détails. Cliquez ensuite sur Terminé.



13. Créez une partition dans le système F5 BIG-IP à configurer et à utiliser avec CIS. Accédez à Système > Utilisateurs > Liste des partitions, cliquez sur Créer et saisissez les détails. Cliquez ensuite sur Terminé.



F5 recommande qu'aucune configuration manuelle ne soit effectuée sur la partition gérée par CIS.

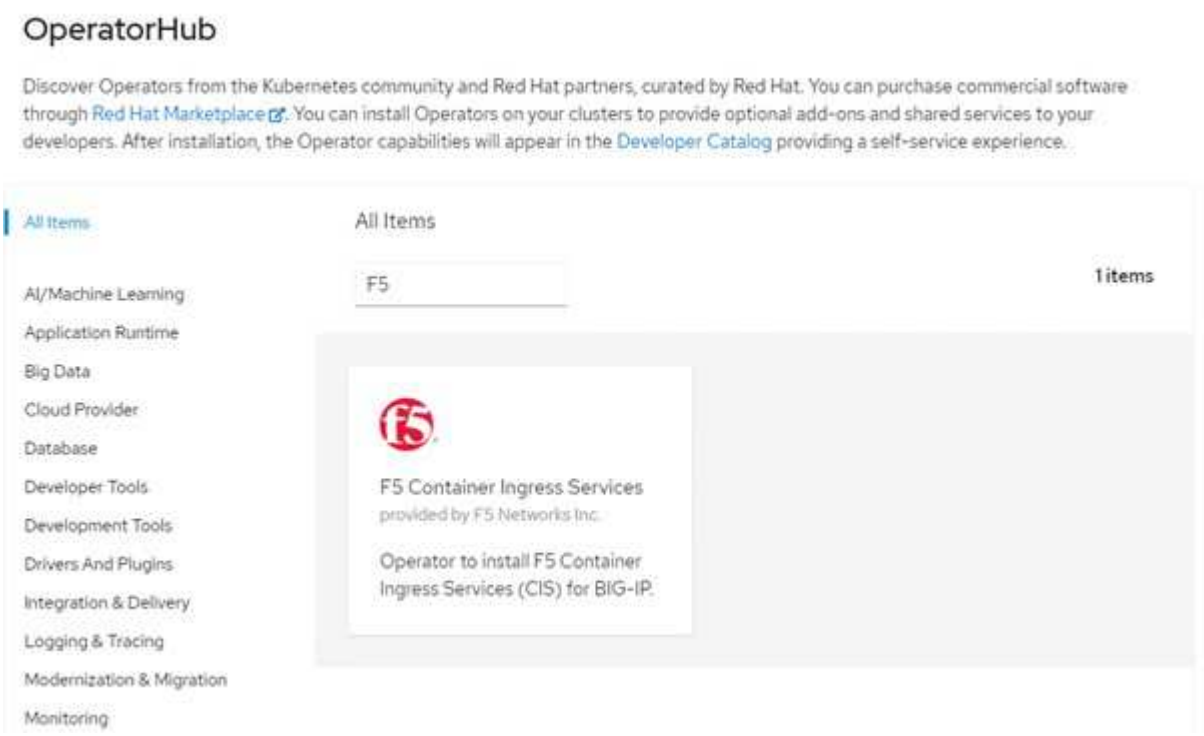
14. Installez le F5 BIG-IP CIS à l'aide de l'opérateur d'OperatorHub. Connectez-vous au cluster Red Hat OpenShift avec les privilèges d'administrateur de cluster et créez un secret avec les informations de connexion au système F5 BIG-IP, ce qui est une condition préalable pour l'opérateur.

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

secret/bigip-login created
```

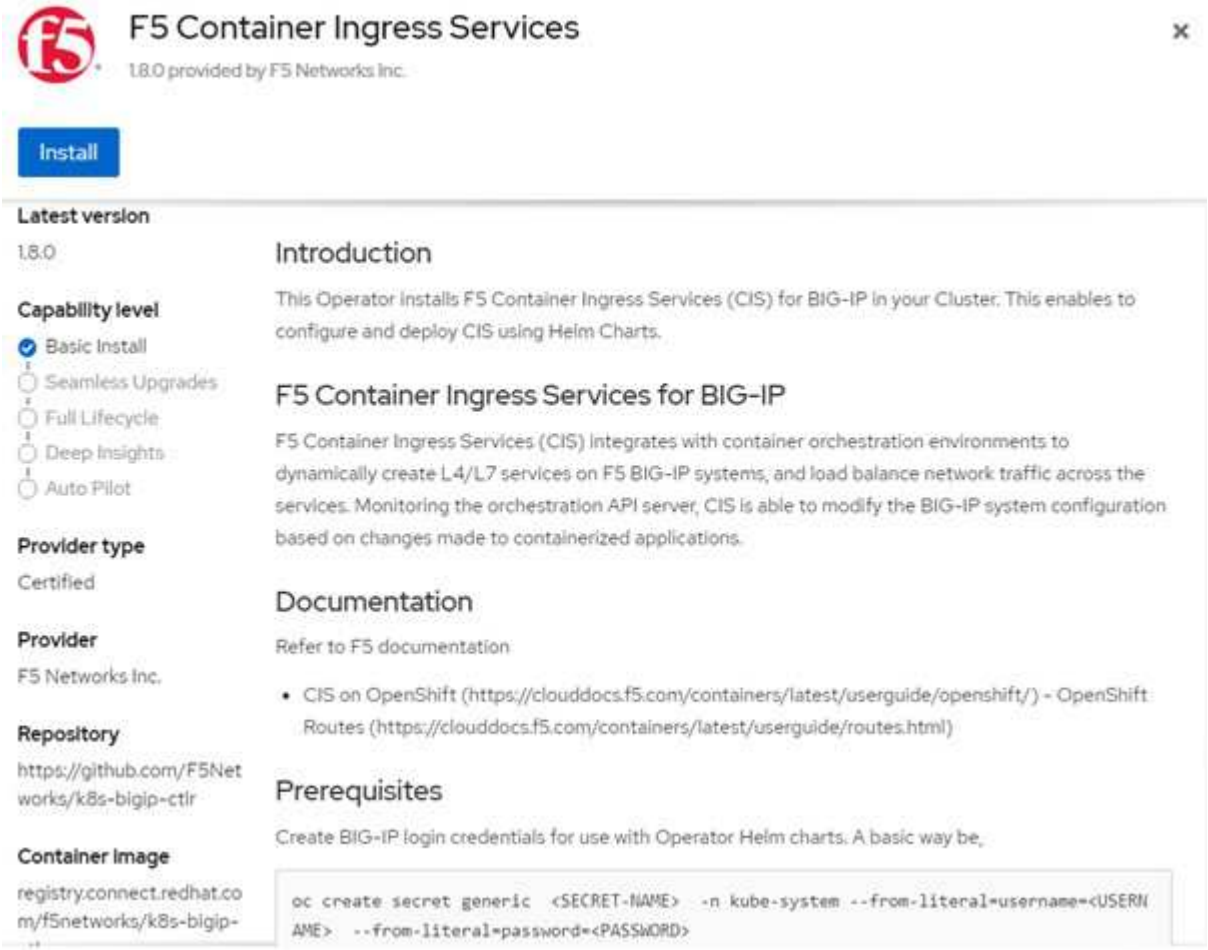
15. Installez les CRD F5 CIS.

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

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

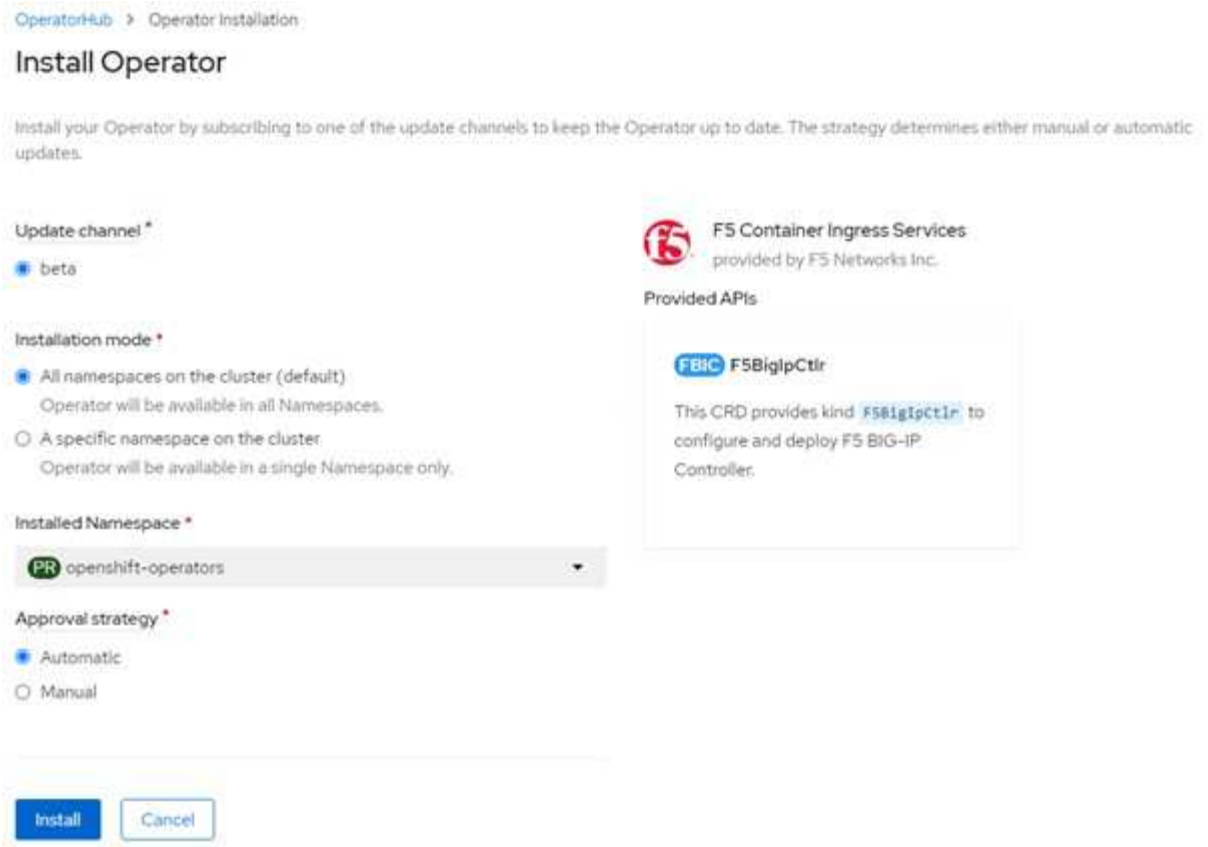
16. Accédez à Opérateurs > OperatorHub, recherchez le mot-clé F5 et cliquez sur la mosaïque Service d'entrée de conteneur F5.



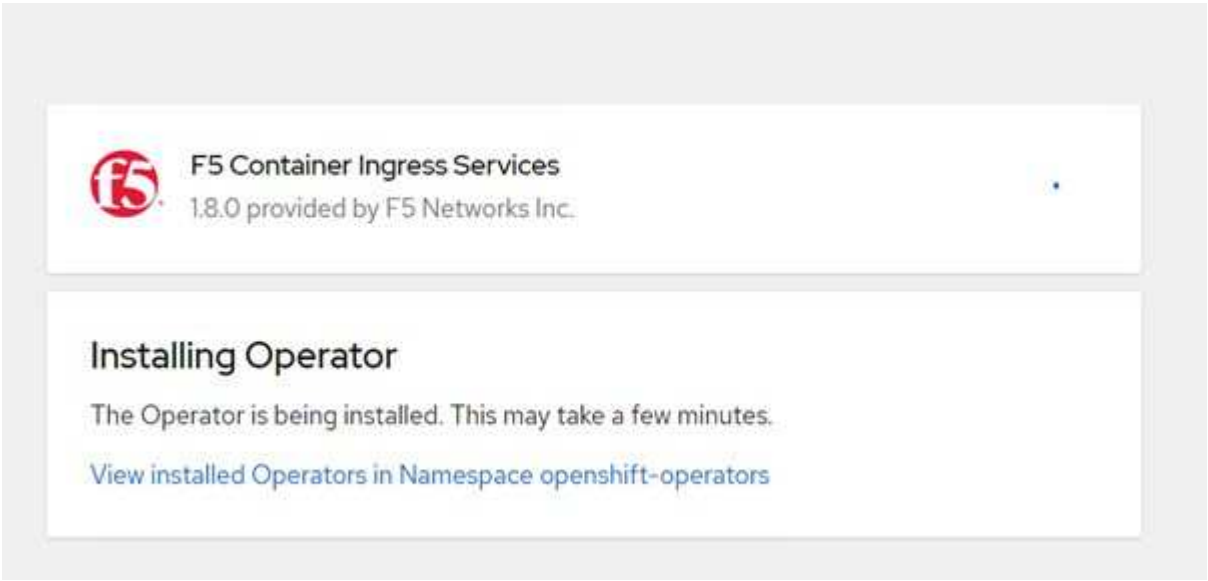
17. Lisez les informations de l'opérateur et cliquez sur Installer.



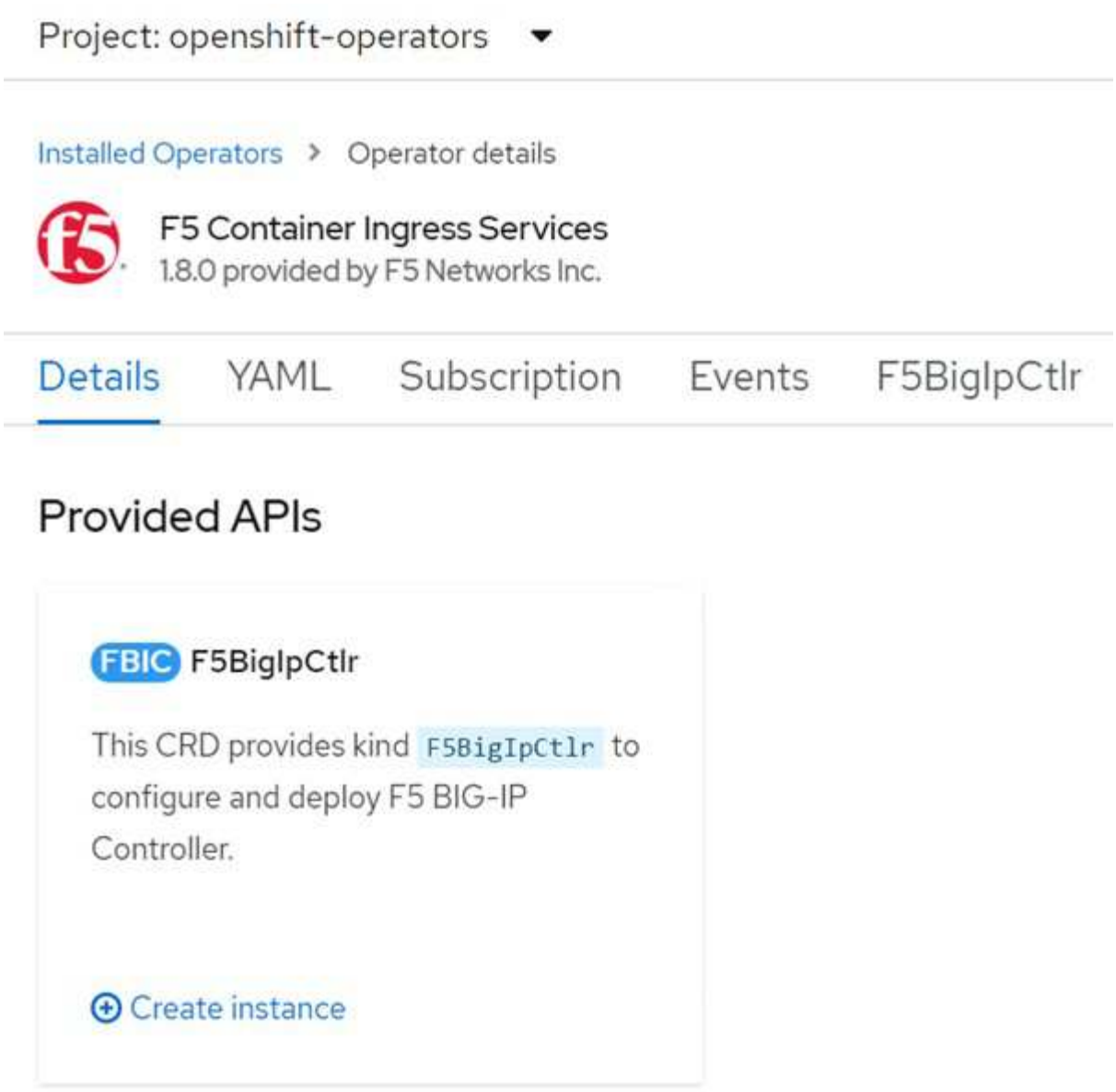
18. Sur l'écran de l'opérateur d'installation, laissez tous les paramètres par défaut et cliquez sur Installer.



19. L'installation de l'opérateur prend un certain temps.



20. Une fois l'opérateur installé, le message Installation réussie s'affiche.
21. Accédez à Opérateurs > Opérateurs installés, cliquez sur F5 Container Ingress Service, puis cliquez sur Créer une instance sous la mosaïque F5BigIpCtlr.



22. Cliquez sur Affichage YAML et collez le contenu suivant après avoir mis à jour les paramètres nécessaires.

> Mettre à jour les paramètres `bigip_partition` , ` openshift_sdn_name`, `bigip_url` et `bigip_login_secret` ci-dessous pour refléter les valeurs de votre configuration avant de copier le contenu.

```
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. Après avoir collé ce contenu, cliquez sur Créer. Cela installe les pods CIS dans l'espace de noms kube-system.

Red Hat OpenShift, par défaut, fournit un moyen d'exposer les services via des routes pour l'équilibrage de charge L7. Un routeur OpenShift intégré est responsable de la publicité et de la gestion du trafic pour ces itinéraires. Cependant, vous pouvez également configurer le F5 CIS pour prendre en charge les routes via un système F5 BIG-IP externe, qui peut fonctionner soit comme un routeur auxiliaire, soit comme un remplacement du routeur OpenShift auto-hébergé. CIS crée un serveur virtuel dans le système BIG-IP qui agit comme un routeur pour les routes OpenShift, et BIG-IP gère la publicité et le routage du trafic. Reportez-vous à la documentation ici pour obtenir des informations sur les paramètres permettant d'activer cette fonctionnalité. Notez que ces paramètres sont définis pour la ressource de déploiement OpenShift dans l'API apps/v1. Par conséquent, lorsque vous les utilisez avec la ressource F5BigIpCtlr cis.f5.com/v1 API, remplacez les tirets (-) par des traits de soulignement (_) pour les noms de paramètres.

24. Les arguments avancés pour la création de ressources CIS incluent `ipam: true` et `custom_resource_mode: true` . Ces paramètres sont requis pour activer l'intégration CIS avec un contrôleur IPAM. Vérifiez que le CIS a activé l'intégration IPAM en créant la ressource IPAM F5.

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

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

25. Créez le compte de service, le rôle et la liaison de rôle requis pour le contrôleur IPAM F5. Créez un fichier YAML et collez le contenu suivant.

```
[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. Créer les ressources.

```
[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. Créez un fichier YAML et collez la définition de déploiement IPAM F5 fournie ci-dessous.

Mettez à jour le paramètre ip-range dans spec.template.spec.containers[0].args ci-dessous pour refléter les ipamLabels et les plages d'adresses IP correspondant à votre configuration.

Étiquettes ipam[`range1` et `range2` [dans l'exemple ci-dessous] doivent être annotés pour les services de type LoadBalancer pour que le contrôleur IPAM détecte et attribue une adresse IP à partir de la plage définie.

```
[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. Créez le déploiement du contrôleur IPAM F5.

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

deployment/f5-ipam-controller created
```

29. Vérifiez que les pods du contrôleur IPAM F5 sont en cours d'exécution.

```
[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. Créez le schéma IPAM F5.

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

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

Vérification

1. Créer un service de type 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. Vérifiez si le contrôleur IPAM lui attribue une IP externe.

```
[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. Créez un déploiement et utilisez le service LoadBalancer qui a été créé.

```
[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. Vérifiez si les pods fonctionnent.

```
[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. Vérifiez si le serveur virtuel correspondant est créé dans le système BIG-IP pour le service de type LoadBalancer dans OpenShift. Accédez à Trafic local > Serveurs virtuels > Liste des serveurs virtuels.

Création de registres d'images privés

Pour la plupart des déploiements de Red Hat OpenShift, l'utilisation d'un registre public comme "Quay.io" ou "DockerHub" répond à la plupart des besoins des clients. Cependant, il arrive parfois qu'un client souhaite héberger ses propres images privées ou personnalisées.

Cette procédure documente la création d'un registre d'images privé soutenu par un volume persistant fourni par Trident et NetApp ONTAP.

> Trident Protect nécessite un registre pour héberger les images requises par les conteneurs Astra . La section suivante décrit les étapes de configuration d'un registre privé sur un cluster Red Hat OpenShift et de transmission des images requises pour prendre en charge l'installation de Trident Protect.

Création d'un registre d'images privé

1. Supprimez l'annotation par défaut de la classe de stockage par défaut actuelle et annotez la classe de stockage basée sur Trident comme valeur par défaut pour le cluster OpenShift.

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

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

2. Modifiez l'opérateur imageregistry en saisissant les paramètres de stockage suivants dans le `spec` section.

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

storage:
  pvc:
    claim:
```

3. Entrez les paramètres suivants dans le `spec` section pour créer une route OpenShift avec un nom d'hôte personnalisé. Enregistrer et quitter.

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

> La configuration d'itinéraire ci-dessus est utilisée lorsque vous souhaitez un nom d'hôte personnalisé pour votre itinéraire. Si vous souhaitez qu'OpenShift crée une route avec un nom d'hôte par défaut, vous pouvez ajouter les paramètres suivants au `spec` section: `defaultRoute: true` .

Certificats TLS personnalisés

Lorsque vous utilisez un nom d'hôte personnalisé pour l'itinéraire, par défaut, il utilise la configuration TLS par défaut de l'opérateur OpenShift Ingress. Cependant, vous pouvez ajouter une configuration TLS personnalisée à l'itinéraire. Pour ce faire, procédez comme suit.

a. Créez un secret avec les certificats TLS et la clé de l'itinéraire.

```
[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. Modifiez l'opérateur imageregistry et ajoutez les paramètres suivants au `spec` section.

```
[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. Modifiez à nouveau l'opérateur imageregistry et changez l'état de gestion de l'opérateur en `Managed` État. Enregistrer et quitter.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Si toutes les conditions préalables sont satisfaites, des PVC, des pods et des services sont créés pour le

registre d’images privé. Dans quelques minutes, le registre devrait être opérationnel.

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

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

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

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

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

```
15h

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

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

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

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

6. Si vous utilisez les certificats TLS par défaut pour la route de registre OpenShift de l'opérateur d'entrée, vous pouvez récupérer les certificats TLS à l'aide de la commande suivante.

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

7. Pour permettre aux nœuds OpenShift d'accéder aux images du registre et de les extraire, ajoutez les certificats au client Docker sur les nœuds OpenShift. Créer une configuration dans le `openshift-config` espace de noms à l'aide des certificats TLS et appliquez-le à la configuration de l'image du cluster pour rendre le certificat fiable.

```
[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. Le registre interne d'OpenShift est contrôlé par authentification. Tous les utilisateurs d'OpenShift peuvent accéder au registre OpenShift, mais les opérations que l'utilisateur connecté peut effectuer dépendent des autorisations de l'utilisateur.
   a. Pour permettre à un utilisateur ou à un groupe d'utilisateurs d'extraire des images du registre, le ou les utilisateurs doivent avoir le rôle de visualiseur de registre attribué.

```
[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. Pour permettre à un utilisateur ou à un groupe d'utilisateurs d'écrire ou de pousser des images, le ou les utilisateurs doivent avoir le rôle d'éditeur de registre attribué.

```
[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. Pour que les nœuds OpenShift puissent accéder au registre et envoyer ou extraire les images, vous devez configurer un secret d'extraction.

```
[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. Ce secret d'extraction peut ensuite être appliqué aux comptes de service ou être référencé dans la définition de pod correspondante.
    a. Pour appliquer le correctif aux comptes de service, exécutez la commande suivante.

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

b. Pour référencer le secret d'extraction dans la définition du pod, ajoutez le paramètre suivant au `spec` section.

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

11. Pour envoyer ou extraire une image à partir de postes de travail autres que le nœud OpenShift, procédez comme suit.

a. Ajoutez les certificats TLS au client Docker.

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

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

b. Connectez-vous à OpenShift à l'aide de la commande oc login.

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

c. Connectez-vous au registre à l'aide des informations d'identification de l'utilisateur OpenShift avec la commande podman/docker.

podman

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

+ REMARQUE : si vous utilisez `kubeadmin` l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu d'un mot de passe.

docker

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

+ REMARQUE : si vous utilisez `kubeadmin` l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu d'un mot de passe.

d. Poussez ou tirez les images.

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
```

Validation de la solution et cas d'utilisation

Validation de la solution et cas d'utilisation : Red Hat OpenShift avec NetApp

Les exemples fournis sur cette page sont des validations de solutions et des cas d'utilisation pour Red Hat OpenShift avec NetApp.

- "Déployer un pipeline Jenkins CI/CD avec stockage persistant"
- "Configurer la multilocation sur Red Hat OpenShift avec NetApp"
- "Virtualisation Red Hat OpenShift avec NetApp ONTAP"
- "Gestion avancée des clusters pour Kubernetes sur Red Hat OpenShift avec NetApp"

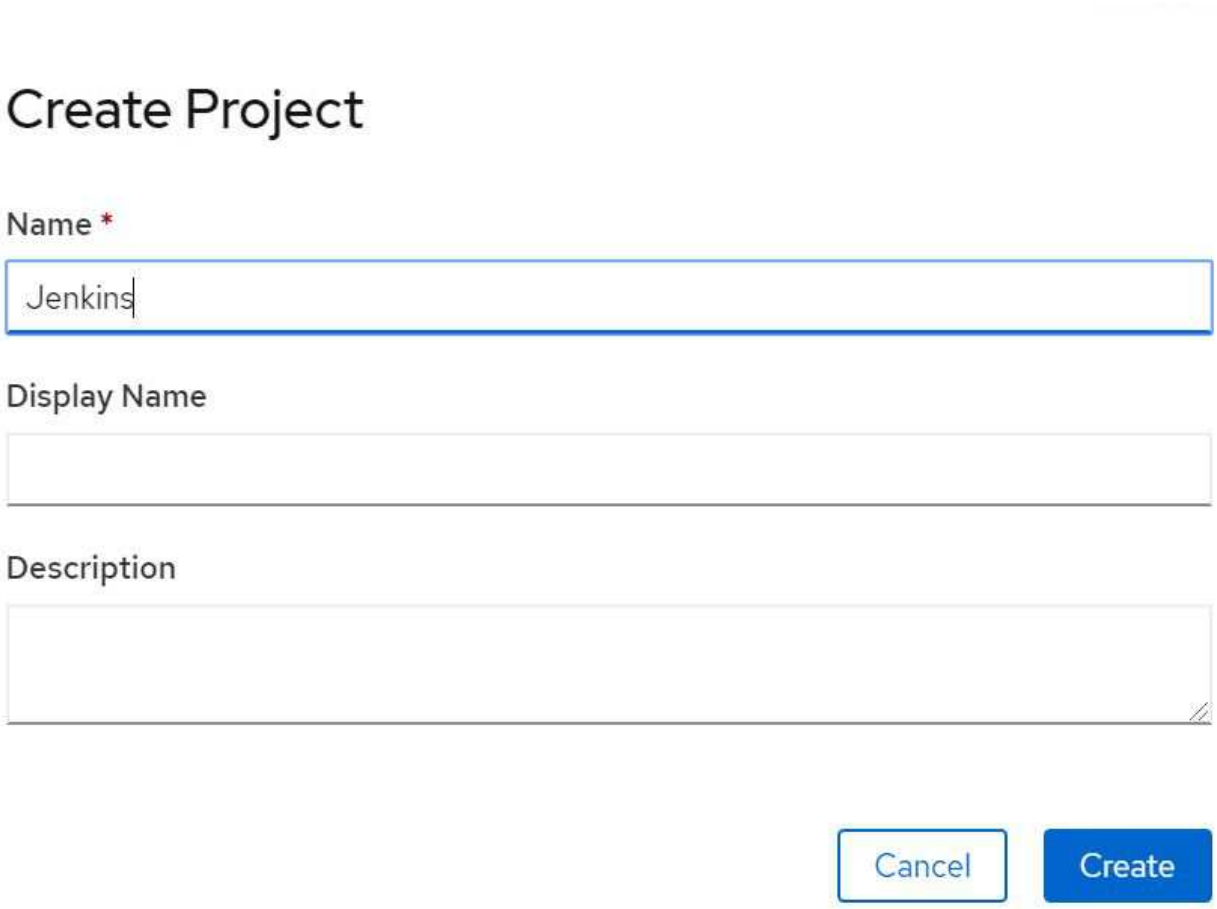
Déployer un pipeline Jenkins CI/CD avec stockage persistant : Red Hat OpenShift avec NetApp

Cette section fournit les étapes à suivre pour déployer un pipeline d'intégration continue/de livraison ou de déploiement continu (CI/CD) avec Jenkins pour valider le fonctionnement de la solution.

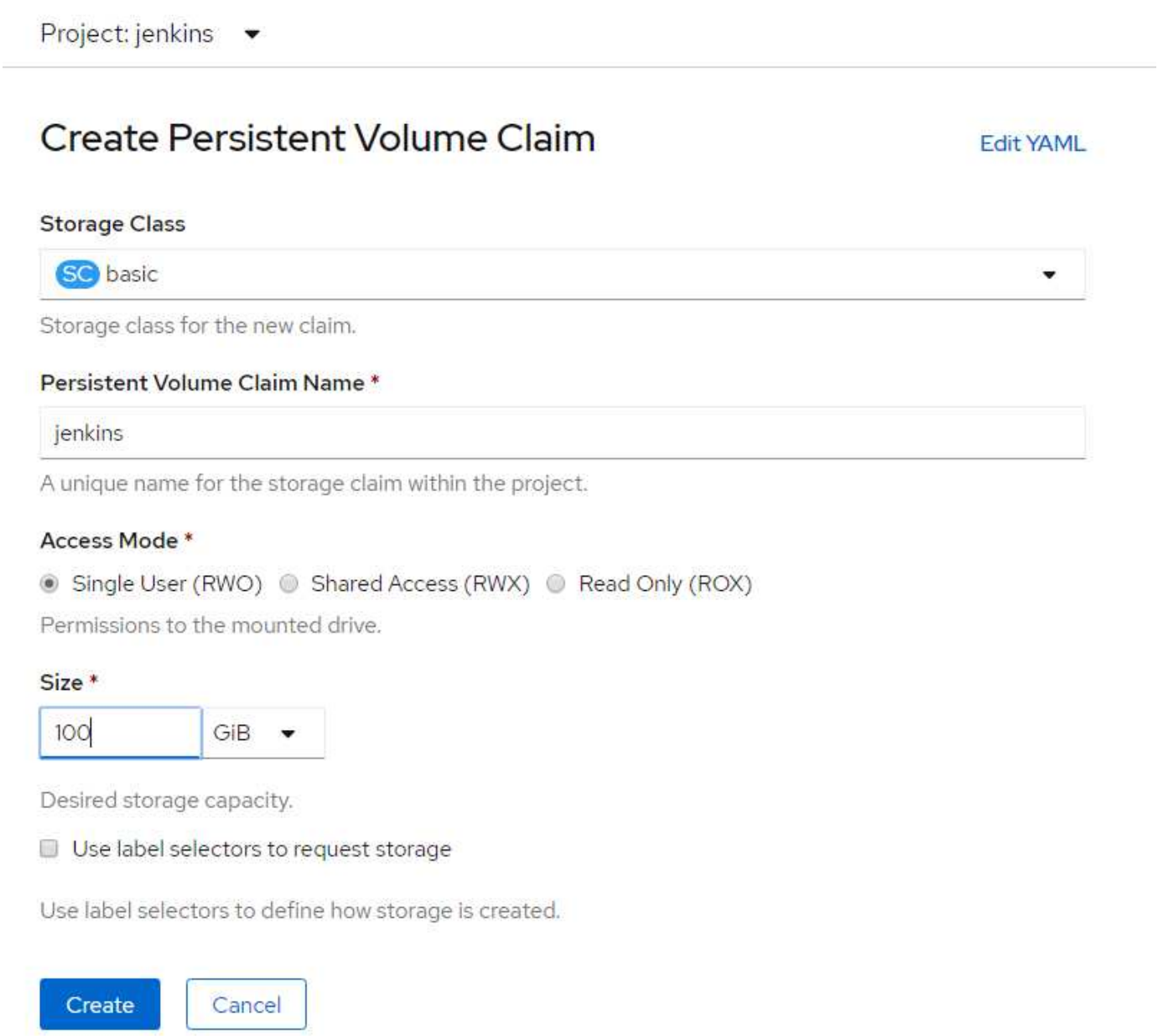
Créer les ressources nécessaires au déploiement de Jenkins

Pour créer les ressources nécessaires au déploiement de l'application Jenkins, procédez comme suit :

1. Créez un nouveau projet nommé Jenkins.



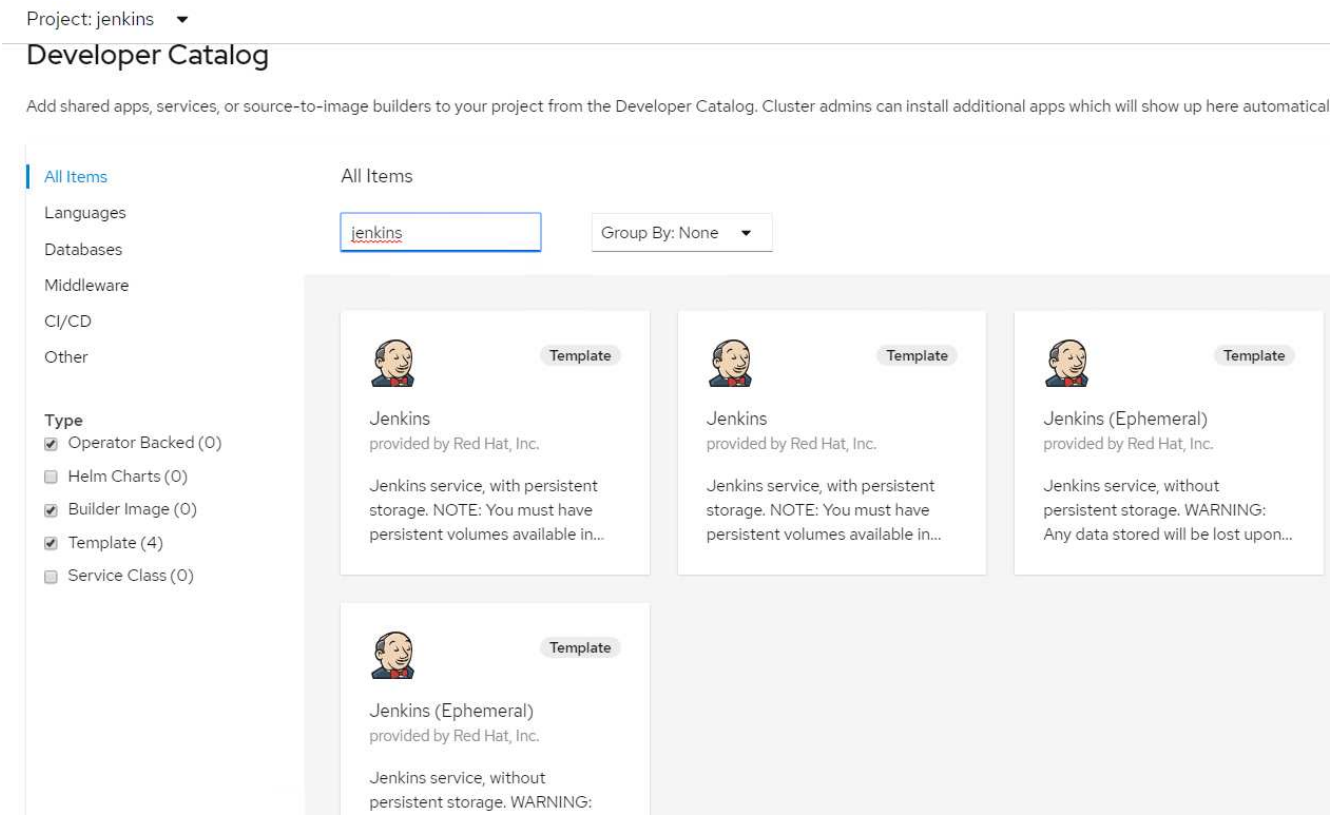
2. Dans cet exemple, nous avons déployé Jenkins avec un stockage persistant. Pour prendre en charge la construction Jenkins, créez le PVC. Accédez à Stockage > Réclamations de volume persistant et cliquez sur Créer une réclamation de volume persistant. Sélectionnez la classe de stockage qui a été créée, assurez-vous que le nom de revendication du volume persistant est jenkins, sélectionnez la taille et le mode d'accès appropriés, puis cliquez sur Créer.



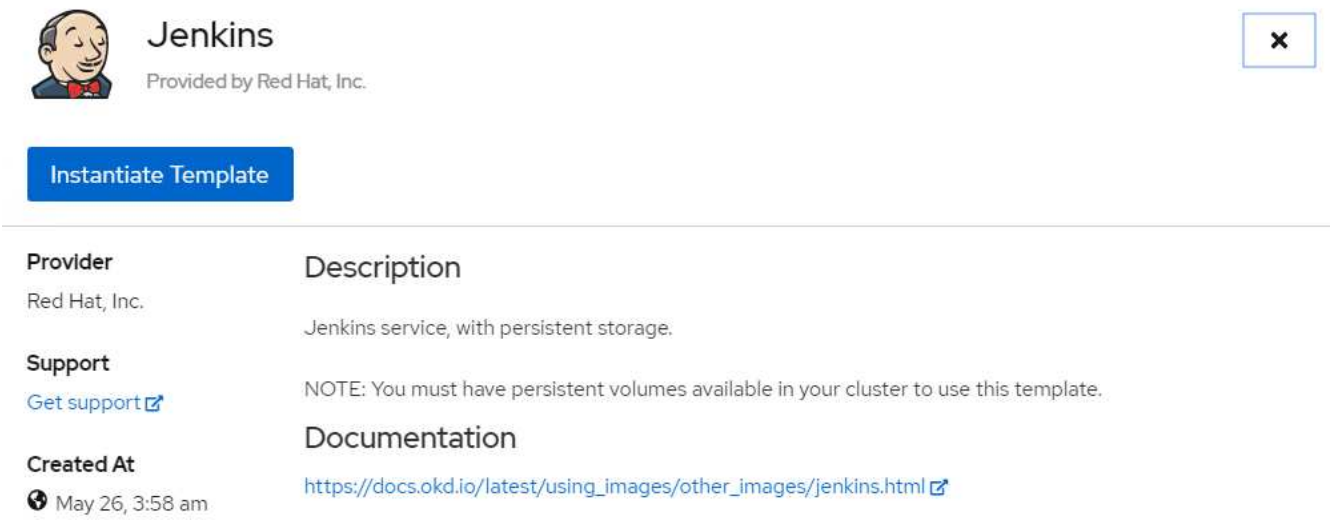
Déployer Jenkins avec un stockage persistant

Pour déployer Jenkins avec un stockage persistant, procédez comme suit :

1. Dans le coin supérieur gauche, changez le rôle d'Administrateur à Développeur. Cliquez sur +Ajouter et sélectionnez À partir du catalogue. Dans la barre Filtrer par mot-clé, recherchez jenkins. Sélectionnez le service Jenkins avec stockage persistant.

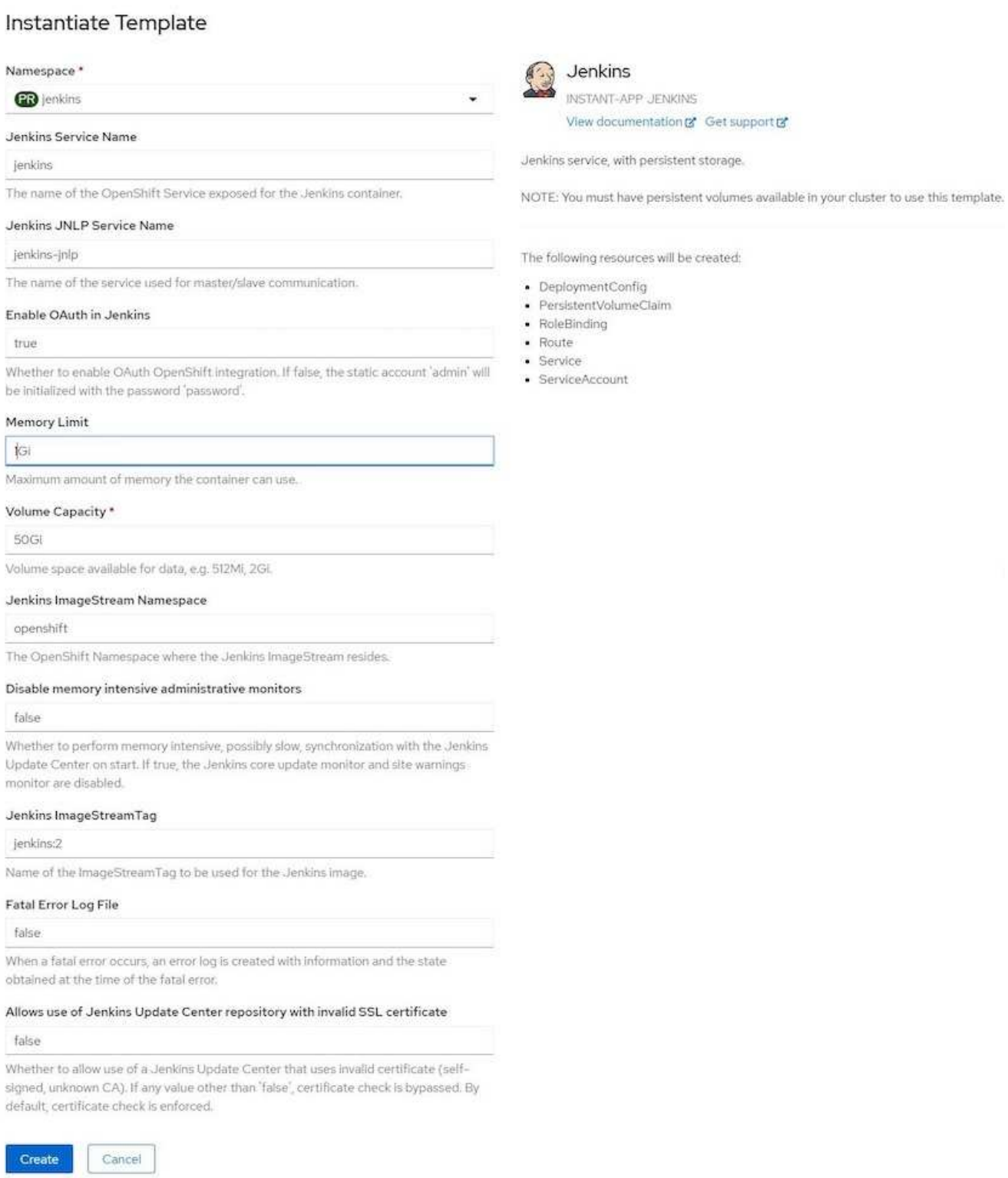


2. Cliquez `Instantiate Template` .

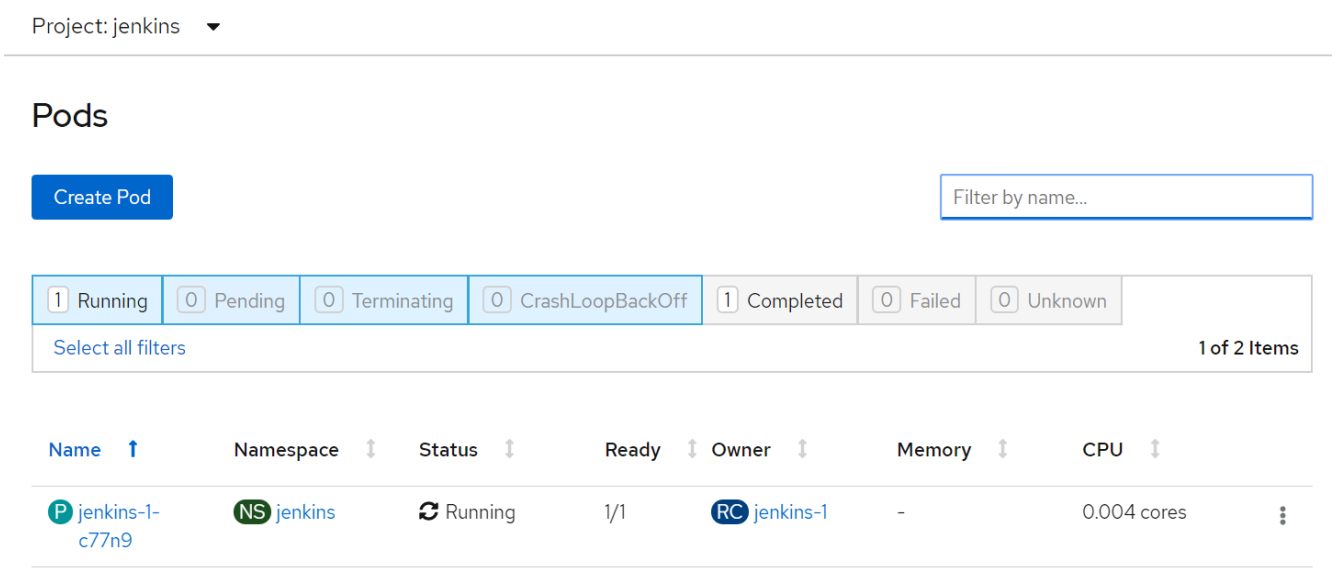


3. Par défaut, les détails de l'application Jenkins sont renseignés. En fonction de vos besoins, modifiez les paramètres et cliquez sur Créer. Ce processus crée toutes les ressources nécessaires pour prendre en

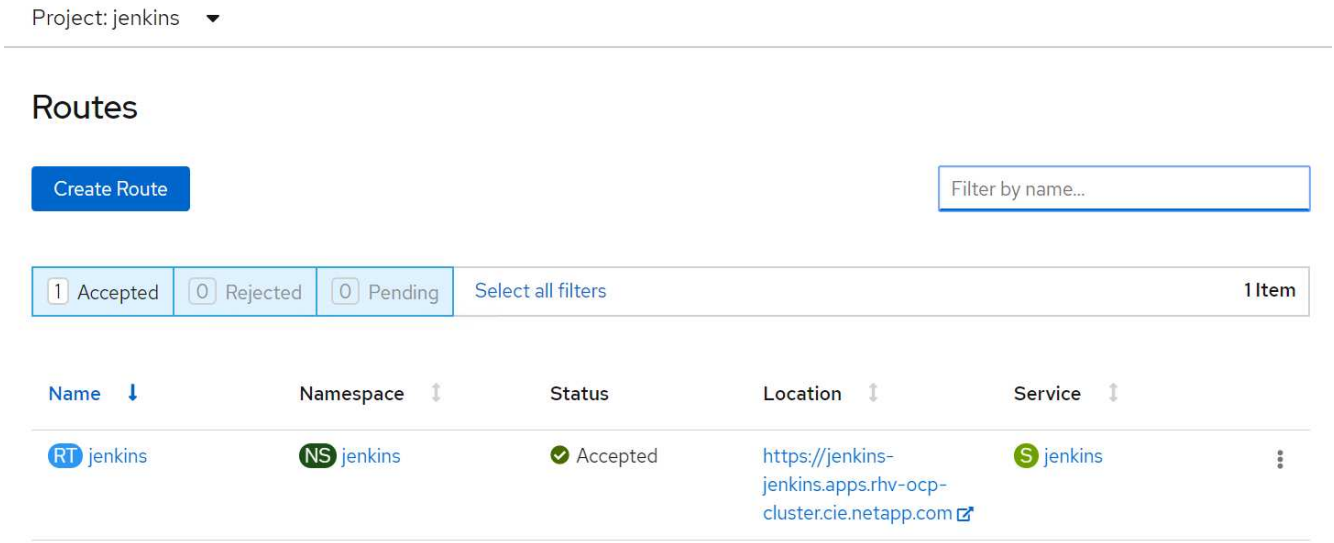
charge Jenkins sur OpenShift.



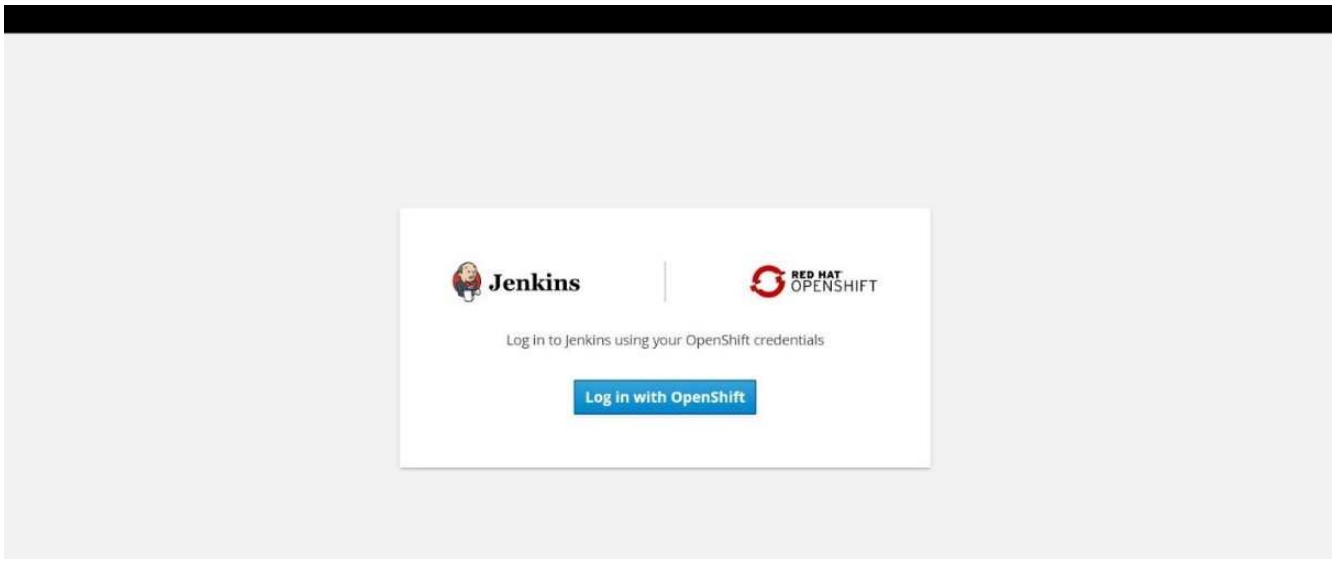
4. Les pods Jenkins prennent environ 10 à 12 minutes pour entrer dans l'état Prêt.



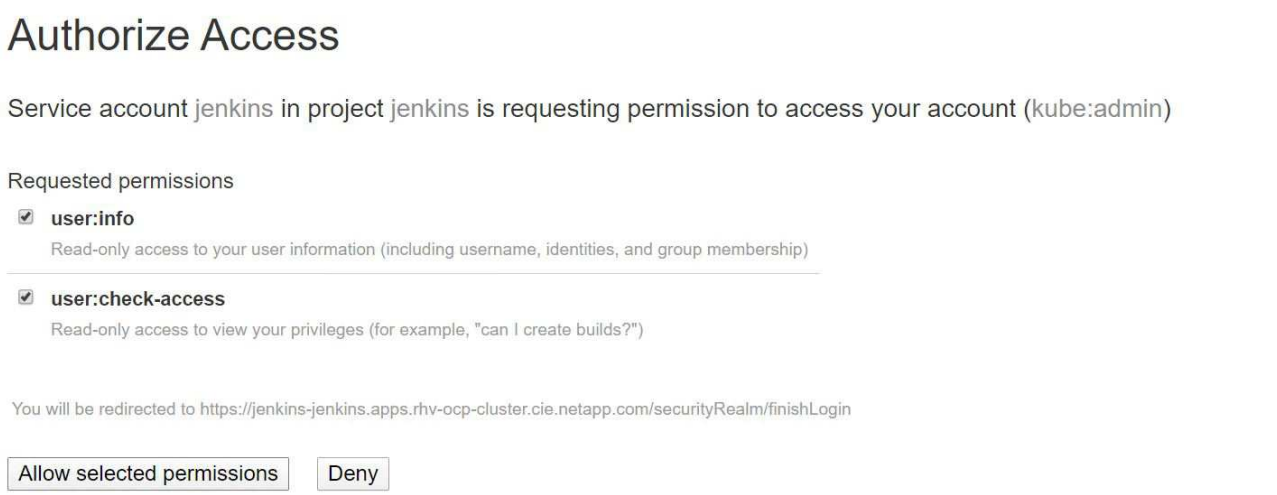
5. Une fois les pods instanciés, accédez à Réseau > Routes. Pour ouvrir la page Web Jenkins, cliquez sur l'URL fournie pour la route Jenkins.



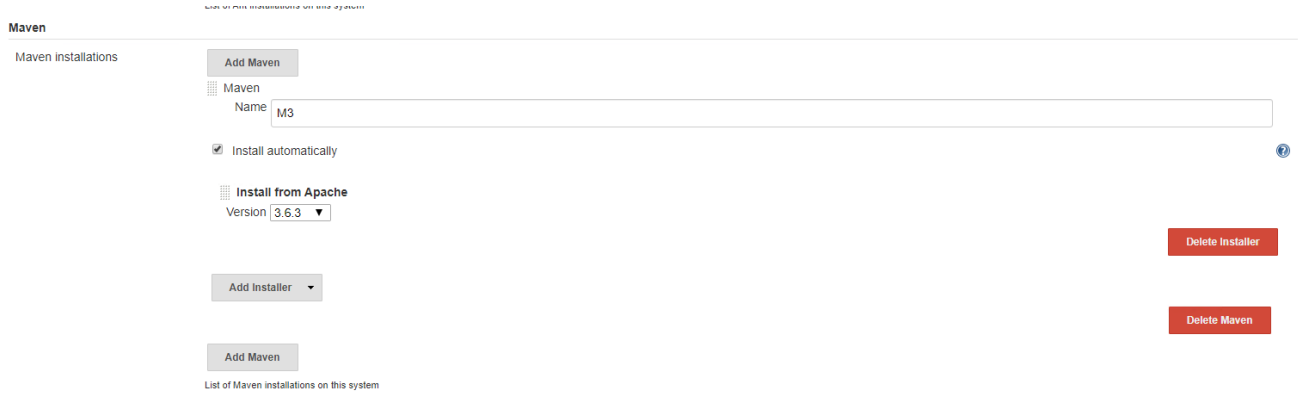
6. Étant donné qu'OpenShift OAuth a été utilisé lors de la création de l'application Jenkins, cliquez sur Se connecter avec OpenShift.



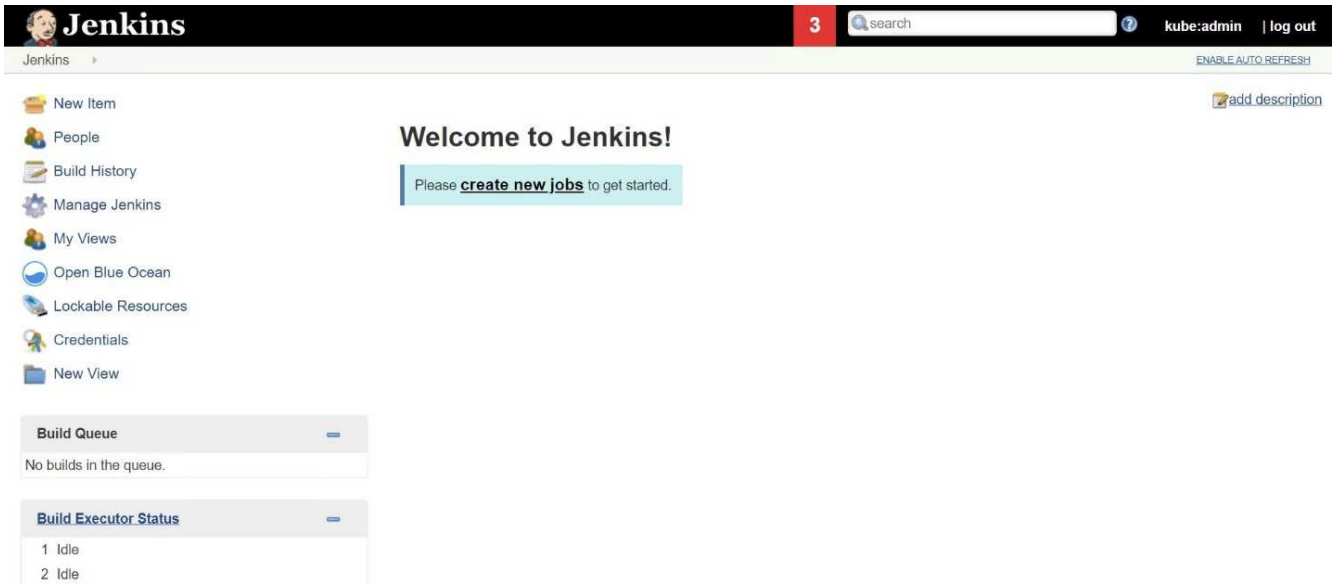
7. Autorisez le compte de service Jenkins à accéder aux utilisateurs OpenShift.



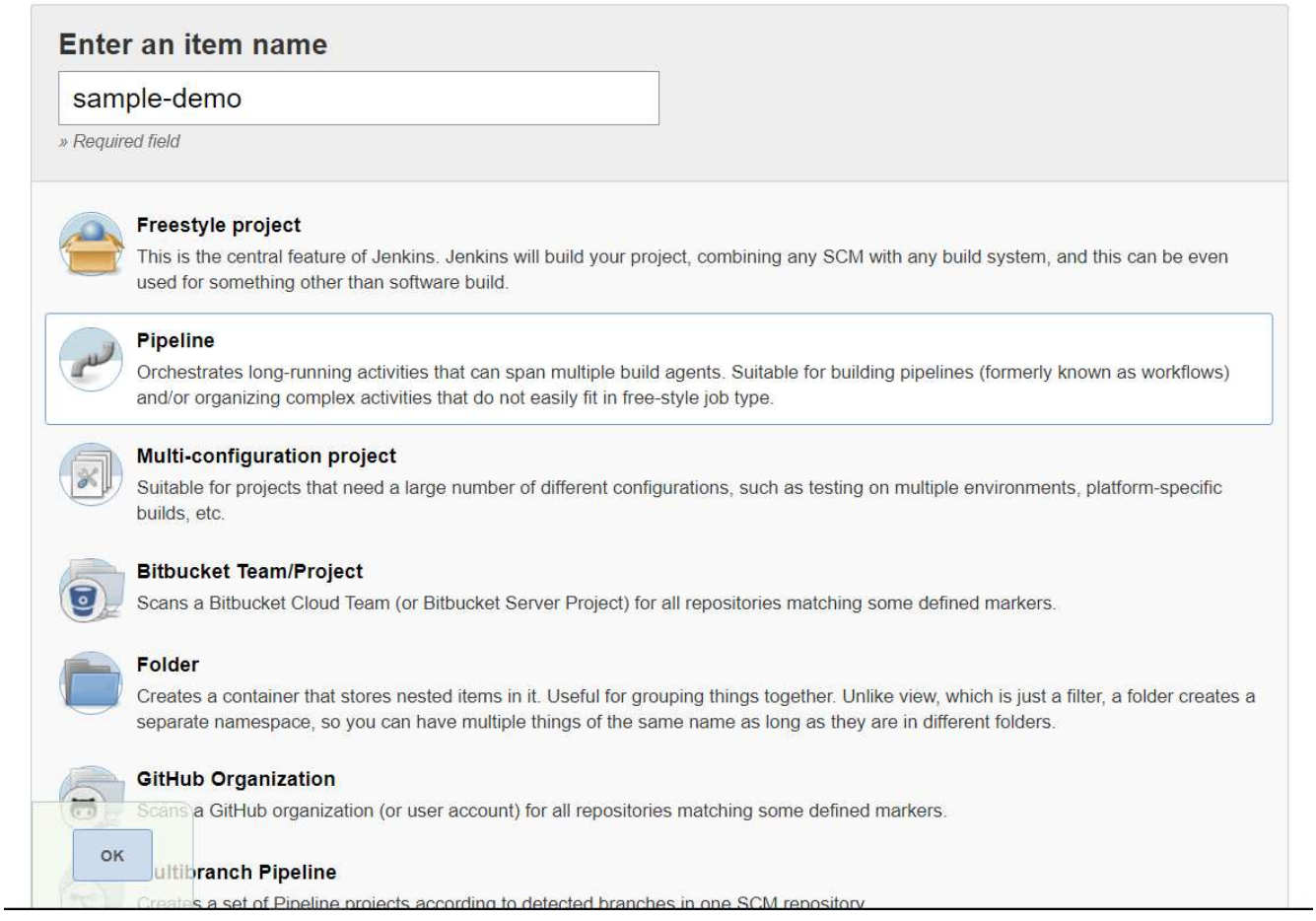
8. La page d'accueil de Jenkins s'affiche. Étant donné que nous utilisons une version Maven, terminez d'abord l'installation de Maven. Accédez à Gérer Jenkins > Configuration globale de l'outil, puis, dans le sous-titre Maven, cliquez sur Ajouter Maven. Entrez le nom de votre choix et assurez-vous que l'option Installer automatiquement est sélectionnée. Cliquez sur Enregistrer.



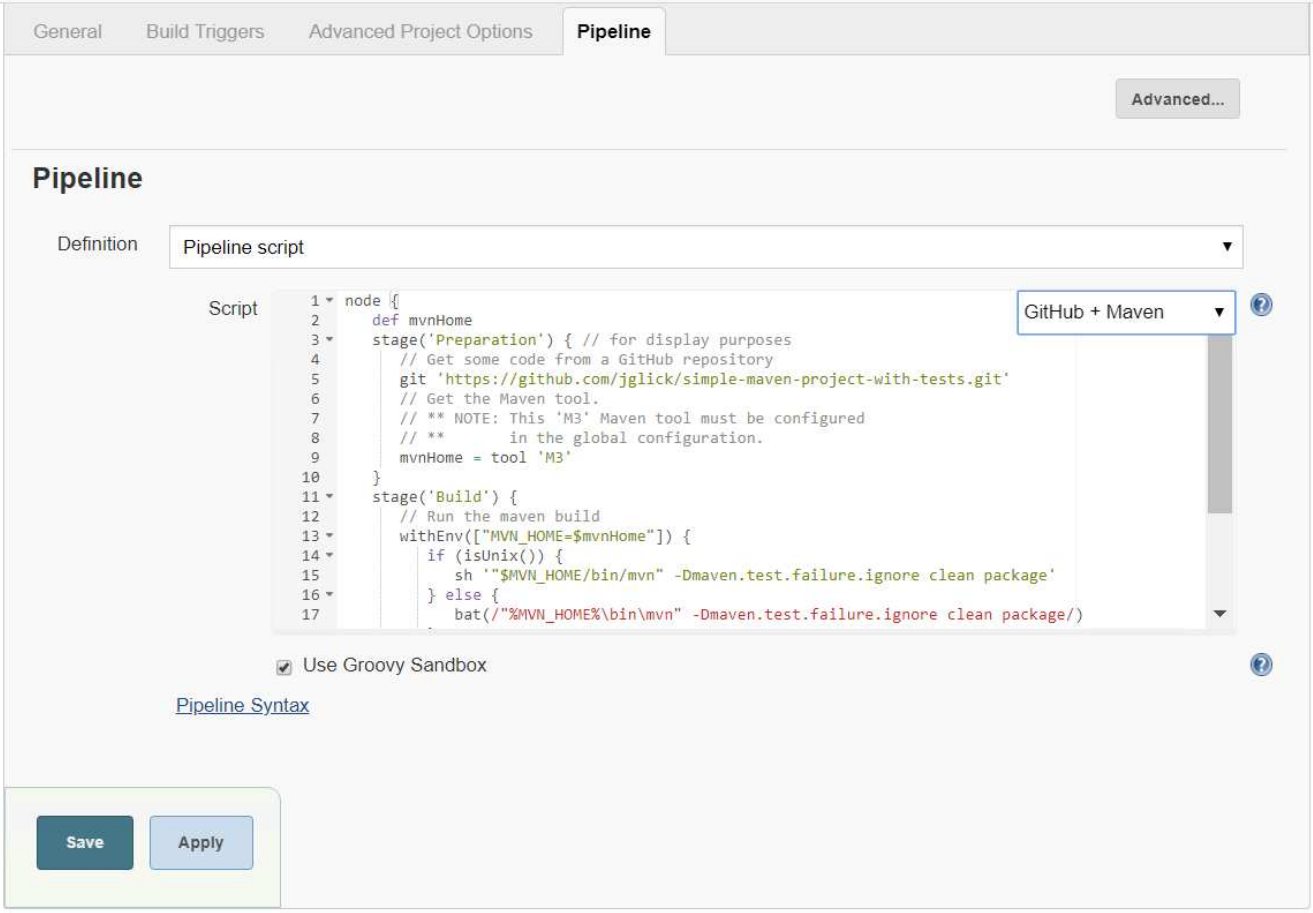
9. Vous pouvez désormais créer un pipeline pour démontrer le flux de travail CI/CD. Sur la page d'accueil, cliquez sur Créer de nouveaux travaux ou sur Nouvel élément dans le menu de gauche.



10. Sur la page Créer un élément, saisissez le nom de votre choix, sélectionnez Pipeline et cliquez sur OK.



11. Sélectionnez l'onglet Pipeline. Dans le menu déroulant Essayer l'exemple de pipeline, sélectionnez Github + Maven. Le code est automatiquement renseigné. Cliquez sur Enregistrer.



12. Cliquez sur Créer maintenant pour déclencher le développement via la phase de préparation, de création et de test. L'achèvement de l'ensemble du processus de construction et l'affichage des résultats de la construction peuvent prendre plusieurs minutes.

13. Chaque fois qu'il y a des modifications de code, le pipeline peut être reconstruit pour corriger la nouvelle version du logiciel, permettant ainsi une intégration et une livraison continues. Cliquez sur Modifications récentes pour suivre les modifications par rapport à la version précédente.

Configurer le multi-tenant

Configuration de la multilocation sur Red Hat OpenShift avec NetApp

De nombreuses organisations qui exécutent plusieurs applications ou charges de travail sur des conteneurs ont tendance à déployer un cluster Red Hat OpenShift par application ou charge de travail. Cela leur permet de mettre en œuvre une isolation stricte pour l'application ou la charge de travail, d'optimiser les performances et de réduire les vulnérabilités de sécurité. Cependant, le déploiement d'un cluster Red Hat OpenShift distinct pour chaque application pose son propre ensemble de problèmes. Cela augmente les frais opérationnels liés à la nécessité de surveiller et de gérer chaque cluster individuellement, augmente les coûts en raison des ressources dédiées à différentes applications et entrave l'évolutivité efficace.

Pour surmonter ces problèmes, on peut envisager d'exécuter toutes les applications ou charges de travail dans un seul cluster Red Hat OpenShift. Mais dans une telle architecture, l'isolement des ressources et les vulnérabilités de sécurité des applications constituent l'un des principaux défis. Toute vulnérabilité de sécurité dans une charge de travail pourrait naturellement se propager à une autre charge de travail, augmentant ainsi la zone d'impact. De plus, toute utilisation brutale et incontrôlée des ressources par une application peut affecter les performances d'une autre application, car il n'existe pas de politique d'allocation des ressources

par défaut.

Par conséquent, les organisations recherchent des solutions qui combinent le meilleur des deux mondes, par exemple en leur permettant d'exécuter toutes leurs charges de travail dans un seul cluster tout en offrant les avantages d'un cluster dédié pour chaque charge de travail.

L'une de ces solutions efficaces consiste à configurer le multi-tenant sur Red Hat OpenShift. La multilocation est une architecture qui permet à plusieurs locataires de coexister sur le même cluster avec une isolation appropriée des ressources, de la sécurité, etc. Dans ce contexte, un locataire peut être considéré comme un sous-ensemble des ressources du cluster configurées pour être utilisées par un groupe particulier d'utilisateurs dans un but exclusif. La configuration de la multilocation sur un cluster Red Hat OpenShift offre les avantages suivants :

- Une réduction des CapEx et des OpEx en permettant le partage des ressources du cluster
- Réduction des frais d'exploitation et de gestion
- Sécuriser les charges de travail contre la contamination croisée des failles de sécurité
- Protection des charges de travail contre une dégradation inattendue des performances due à une contention des ressources

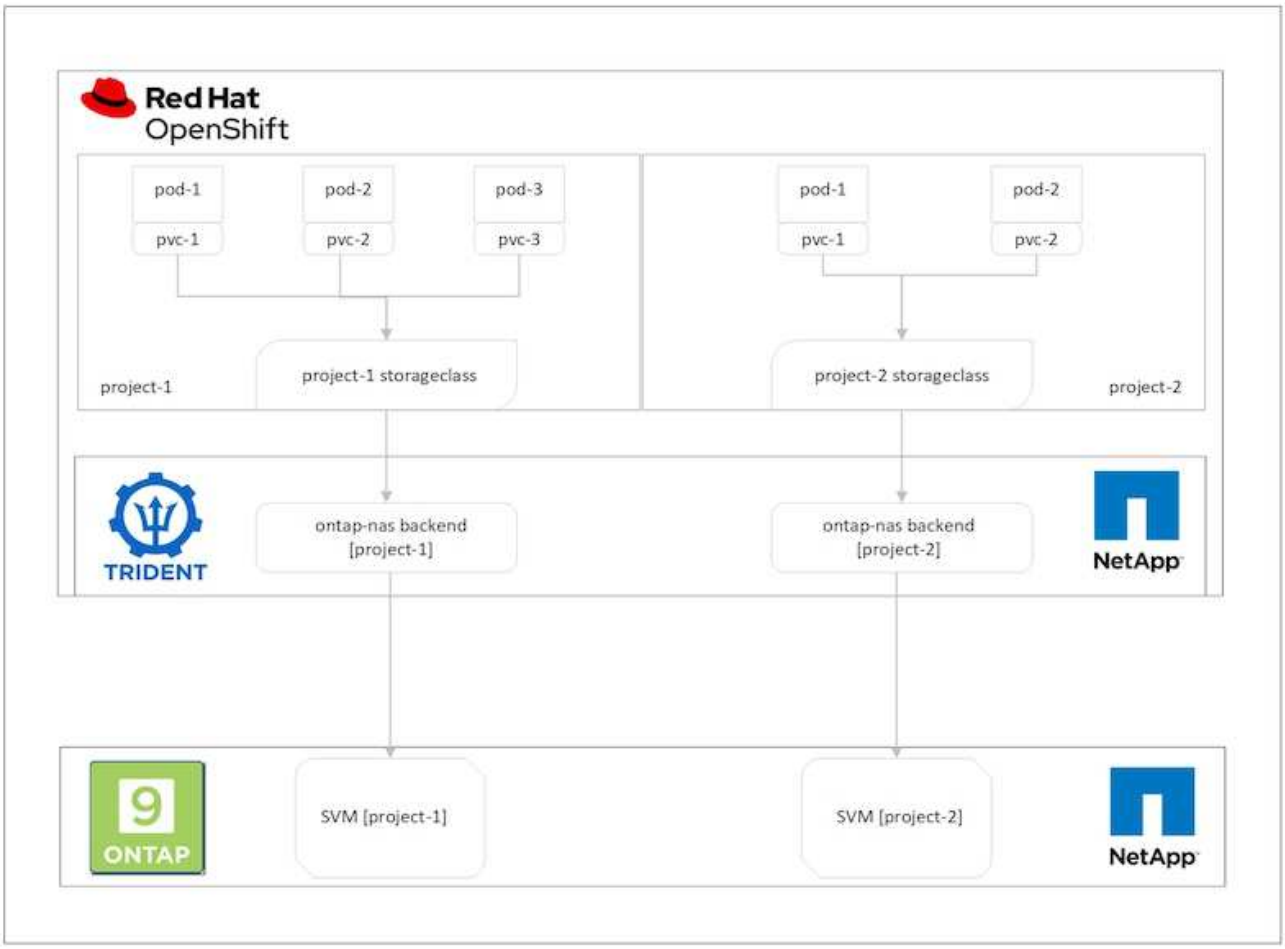
Pour un cluster OpenShift multilocataire entièrement réalisé, des quotas et des restrictions doivent être configurés pour les ressources du cluster appartenant à différents compartiments de ressources : calcul, stockage, réseau, sécurité, etc. Bien que nous couvrions certains aspects de tous les compartiments de ressources de cette solution, nous nous concentrons sur les meilleures pratiques pour isoler et sécuriser les données servies ou consommées par plusieurs charges de travail sur le même cluster Red Hat OpenShift en configurant la multilocation sur les ressources de stockage qui sont allouées dynamiquement par Trident soutenu par NetApp ONTAP.

Architecture

Bien que Red Hat OpenShift et Trident soutenus par NetApp ONTAP ne fournissent pas d'isolation entre les charges de travail par défaut, ils offrent une large gamme de fonctionnalités qui peuvent être utilisées pour configurer la multilocation. Pour mieux comprendre la conception d'une solution multilocataire sur un cluster Red Hat OpenShift avec Trident soutenu par NetApp ONTAP, considérons un exemple avec un ensemble d'exigences et décrivons la configuration qui l'entoure.

Supposons qu'une organisation exécute deux de ses charges de travail sur un cluster Red Hat OpenShift dans le cadre de deux projets sur lesquels travaillent deux équipes différentes. Les données de ces charges de travail résident sur des PVC provisionnés dynamiquement par Trident sur un backend NAS NetApp ONTAP . L'organisation a besoin de concevoir une solution multilocataire pour ces deux charges de travail et d'isoler les ressources utilisées pour ces projets afin de garantir que la sécurité et les performances sont maintenues, principalement axées sur les données qui servent ces applications.

La figure suivante illustre la solution multilocataire sur un cluster Red Hat OpenShift avec Trident soutenu par NetApp ONTAP.



Exigences technologiques

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

Red Hat OpenShift – Ressources de cluster

Du point de vue du cluster Red Hat OpenShift, la ressource de niveau supérieur par laquelle commencer est le projet. Un projet OpenShift peut être considéré comme une ressource de cluster qui divise l'ensemble du cluster OpenShift en plusieurs clusters virtuels. Par conséquent, l'isolement au niveau du projet fournit une base pour la configuration de la multilocation.

L'étape suivante consiste à configurer RBAC dans le cluster. La meilleure pratique consiste à configurer tous les développeurs travaillant sur un seul projet ou une seule charge de travail dans un seul groupe d'utilisateurs dans le fournisseur d'identité (IdP). Red Hat OpenShift permet l'intégration IdP et la synchronisation des groupes d'utilisateurs, permettant ainsi aux utilisateurs et aux groupes de l'IdP d'être importés dans le cluster. Cela aide les administrateurs de cluster à séparer l'accès aux ressources de cluster dédiées à un projet à un ou plusieurs groupes d'utilisateurs travaillant sur ce projet, limitant ainsi l'accès non autorisé à toutes les ressources de cluster. Pour en savoir plus sur l'intégration d'IdP avec Red Hat OpenShift, consultez la documentation "ici" .

NetApp ONTAP

Il est important d'isoler le stockage partagé servant de fournisseur de stockage persistant pour un cluster Red Hat OpenShift afin de garantir que les volumes créés sur le stockage pour chaque projet apparaissent aux hôtes comme s'ils étaient créés sur un stockage séparé. Pour ce faire, créez autant de SVM (machines virtuelles de stockage) sur NetApp ONTAP qu'il y a de projets ou de charges de travail, et dédiez chaque SVM à une charge de travail.

Trident

Une fois que vous avez créé différents SVM pour différents projets sur NetApp ONTAP, vous devez mapper chaque SVM à un backend Trident différent. La configuration du backend sur Trident pilote l'allocation du stockage persistant aux ressources du cluster OpenShift et nécessite que les détails du SVM soient mappés. Cela devrait être au minimum le pilote de protocole pour le backend. En option, il vous permet de définir comment les volumes sont provisionnés sur le stockage et de définir des limites pour la taille des volumes ou l'utilisation des agrégats, etc. Les détails concernant la définition des backends Trident peuvent être trouvés "ici" .

Red Hat OpenShift – ressources de stockage

Après avoir configuré les backends Trident , l'étape suivante consiste à configurer les StorageClasses. Configurez autant de classes de stockage qu'il y a de backends, en fournissant à chaque classe de stockage l'accès pour faire tourner des volumes uniquement sur un backend. Nous pouvons mapper la StorageClass à un backend Trident particulier en utilisant le paramètre storagePools lors de la définition de la classe de stockage. Les détails pour définir une classe de stockage peuvent être trouvés "ici" . Il existe donc un mappage un à un de StorageClass vers le backend Trident qui pointe vers un SVM. Cela garantit que toutes les demandes de stockage via la StorageClass attribuée à ce projet sont traitées par le SVM dédié à ce projet uniquement.

Étant donné que les classes de stockage ne sont pas des ressources d'espace de noms, comment pouvons-nous garantir que les revendications de stockage sur la classe de stockage d'un projet par des pods dans un autre espace de noms ou projet soient rejetées ? La réponse est d'utiliser ResourceQuotas. Les ResourceQuotas sont des objets qui contrôlent l'utilisation totale des ressources par projet. Il peut limiter le nombre ainsi que la quantité totale de ressources pouvant être consommées par les objets du projet. Presque toutes les ressources d'un projet peuvent être limitées à l'aide de ResourceQuotas et leur utilisation efficace peut aider les organisations à réduire les coûts et les pannes dues au surprovisionnement ou à la surconsommation de ressources. Se référer à la documentation "ici" pour plus d'informations.

Pour ce cas d'utilisation, nous devons empêcher les pods d'un projet particulier de réclamer du stockage à partir de classes de stockage qui ne sont pas dédiées à leur projet. Pour ce faire, nous devons limiter les revendications de volume persistant pour d'autres classes de stockage en définissant `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` à 0. De plus, un administrateur de cluster doit s'assurer que les développeurs d'un projet ne doivent pas avoir accès à la modification des ResourceQuotas.

Configuration

Pour toute solution multilocataire, aucun utilisateur ne peut avoir accès à plus de ressources de cluster que nécessaire. Ainsi, l'ensemble des ressources à configurer dans le cadre de la configuration multi-location est divisé entre l'administrateur du cluster, l'administrateur du stockage et les développeurs travaillant sur chaque projet.

Le tableau suivant décrit les différentes tâches à effectuer par les différents utilisateurs :

Rôle	Tâches
Administrateur de cluster	Créer des projets pour différentes applications ou charges de travail
	Créer des ClusterRoles et des RoleBindings pour storage-admin
	Créer des rôles et des liaisons de rôle pour les développeurs attribuant l'accès à des projets spécifiques
	[Facultatif] Configurer des projets pour planifier des pods sur des nœuds spécifiques
Administrateur de stockage	Créer des SVM sur NetApp ONTAP
	Créer des backends Trident
	Créer des StorageClasses
	Créer des quotas de ressources de stockage
Développeurs	Valider l'accès pour créer ou corriger des PVC ou des pods dans le projet attribué
	Valider l'accès pour créer ou corriger des PVC ou des pods dans un autre projet
	Valider l'accès pour afficher ou modifier les projets, les quotas de ressources et les classes de stockage

Configuration

Voici les conditions préalables à la configuration de la multilocation sur Red Hat OpenShift avec NetApp.

Prérequis

- Cluster NetApp ONTAP
- Cluster Red Hat OpenShift
- Trident installé sur le cluster
- Poste de travail d'administration avec les outils tridentctl et oc installés et ajoutés à $PATH
- Accès administrateur à ONTAP
- Accès administrateur du cluster au cluster OpenShift
- Le cluster est intégré au fournisseur d'identité
- Le fournisseur d'identité est configuré pour distinguer efficacement les utilisateurs de différentes équipes

Configuration : tâches d'administration du cluster

Les tâches suivantes sont effectuées par l'administrateur du cluster Red Hat OpenShift :

1. Connectez-vous au cluster Red Hat OpenShift en tant qu'administrateur du cluster.
2. Créez deux projets correspondant à des projets différents.

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

3. Créez le rôle de développeur pour le projet-1.

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

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

La définition de rôle fournie dans cette section n'est qu'un exemple. Les rôles des développeurs doivent être définis en fonction des besoins des utilisateurs finaux.

1. De même, créez des rôles de développeur pour le projet 2.
2. Toutes les ressources de stockage OpenShift et NetApp sont généralement gérées par un administrateur de stockage. L'accès des administrateurs de stockage est contrôlé par le rôle d'opérateur Trident créé lors de l'installation de Trident . En plus de cela, l'administrateur du stockage a également besoin d'accéder à ResourceQuotas pour contrôler la manière dont le stockage est consommé.
3. Créez un rôle pour gérer les ResourceQuotas dans tous les projets du cluster afin de l'attacher à l'administrateur de stockage.

```
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. Assurez-vous que le cluster est intégré au fournisseur d'identité de l'organisation et que les groupes d'utilisateurs sont synchronisés avec les groupes de cluster. L'exemple suivant montre que le fournisseur d'identité a été intégré au cluster et synchronisé avec les groupes d'utilisateurs.

```
$ 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. Configurez ClusterRoleBindings pour les administrateurs de stockage.

```
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
```

Pour les administrateurs de stockage, deux rôles doivent être liés : trident-operator et resource-quotas.

1. Créez des RoleBindings pour les développeurs liant le rôle developer-project-1 au groupe correspondant (ocp-project-1) dans project-1.

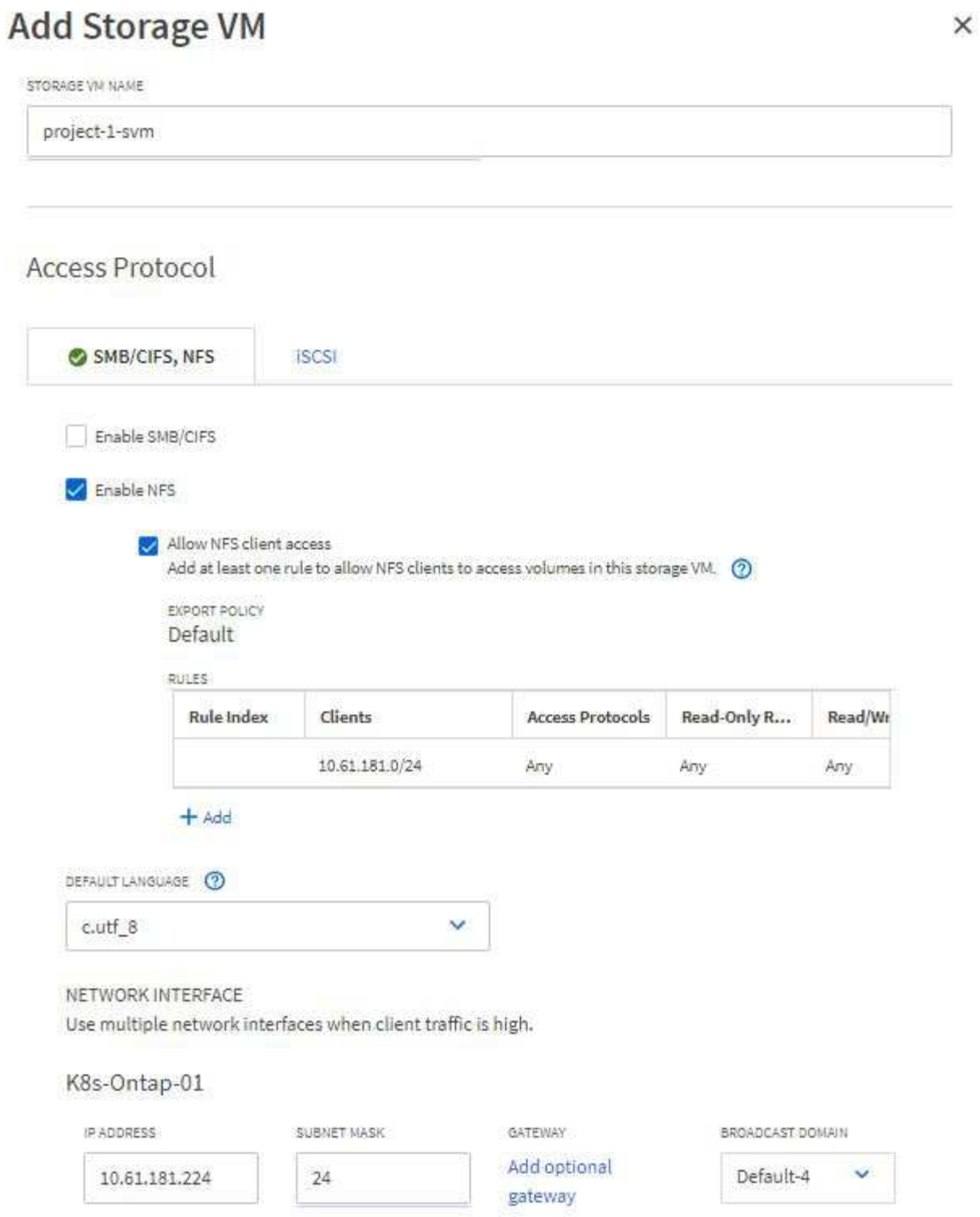
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role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

2. De même, créez des RoleBindings pour les développeurs liant les rôles de développeur au groupe d'utilisateurs correspondant dans le projet-2.

Configuration : tâches d'administration du stockage

Les ressources suivantes doivent être configurées par un administrateur de stockage :

1. Connectez-vous au cluster NetApp ONTAP en tant qu'administrateur.
2. Accédez à Stockage > Machines virtuelles de stockage et cliquez sur Ajouter. Créez deux SVM, un pour le projet 1 et l'autre pour le projet 2, en fournissant les détails requis. Créez également un compte vsadmin pour gérer le SVM et ses ressources.



1. Connectez-vous au cluster Red Hat OpenShift en tant qu'administrateur de stockage.
2. Créez le backend pour le projet-1 et mappez-le au SVM dédié au projet. NetApp recommande d'utiliser le compte vsadmin du SVM pour connecter le backend au SVM au lieu d'utiliser l'administrateur du cluster ONTAP .

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

Nous utilisons le pilote ontap-nas pour cet exemple. Utilisez le pilote approprié lors de la création du backend en fonction du cas d'utilisation.

Nous supposons que Trident est installé dans le projet Trident.

1. De même, créez le backend Trident pour le projet-2 et mappez-le au SVM dédié au projet-2.
2. Ensuite, créez les classes de stockage. Créez la classe de stockage pour le projet-1 et configurez-la pour utiliser les pools de stockage du backend dédié au projet-1 en définissant le paramètre storagePools.

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

3. De même, créez une classe de stockage pour le projet-2 et configurez-la pour utiliser les pools de stockage du backend dédié au projet-2.
4. Créez un ResourceQuota pour restreindre les ressources du projet 1 demandant du stockage à partir de classes de stockage dédiées à d'autres projets.

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

5. De même, créez un ResourceQuota pour restreindre les ressources du projet 2 demandant du stockage à partir de classes de stockage dédiées à d'autres projets.

Validation

Pour valider l'architecture multilocataire configurée aux étapes précédentes, procédez comme suit :

Valider l'accès pour créer des PVC ou des pods dans le projet attribué

1. Connectez-vous en tant qu'utilisateur ocp-project-1, développeur dans le projet-1.
2. Vérifiez l'accès pour créer un nouveau projet.

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

3. Créez un PVC dans le projet-1 en utilisant la classe de stockage attribuée au projet-1.

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

4. Vérifiez le PV associé au PVC.

```
oc get pv
```

5. Validez que le PV et son volume sont créés dans une SVM dédiée au projet-1 sur NetApp ONTAP.

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

6. Créez un pod dans le projet-1 et montez le PVC créé à l'étape précédente.

```
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. Vérifiez si le pod est en cours d'exécution et s'il a monté le volume.

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

Valider l'accès pour créer des PVC ou des pods dans un autre projet ou utiliser des ressources dédiées à un autre projet

1. Connectez-vous en tant qu'utilisateur ocp-project-1, développeur dans le projet-1.
2. Créez un PVC dans le projet-1 en utilisant la classe de stockage attribuée au projet-2.

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

3. Créez un PVC dans le projet-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. Assurez-vous que les PVC `test-pvc-project-1-sc-2` et `test-pvc-project-2-sc-1` n'ont pas été créés.

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

5. Créez un pod dans le projet-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
```

Valider l'accès pour afficher et modifier les projets, les quotas de ressources et les classes de stockage

1. Connectez-vous en tant qu'utilisateur ocp-project-1, développeur dans le projet-1.
2. Vérifiez l'accès pour créer de nouveaux projets.

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

3. Valider l'accès pour visualiser les projets.

```
oc get ns
```

4. Vérifiez si l'utilisateur peut afficher ou modifier les ResourceQuotas dans le projet-1.

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

5. Validez que l'utilisateur a accès pour afficher les classes de stockage.

```
oc get sc
```

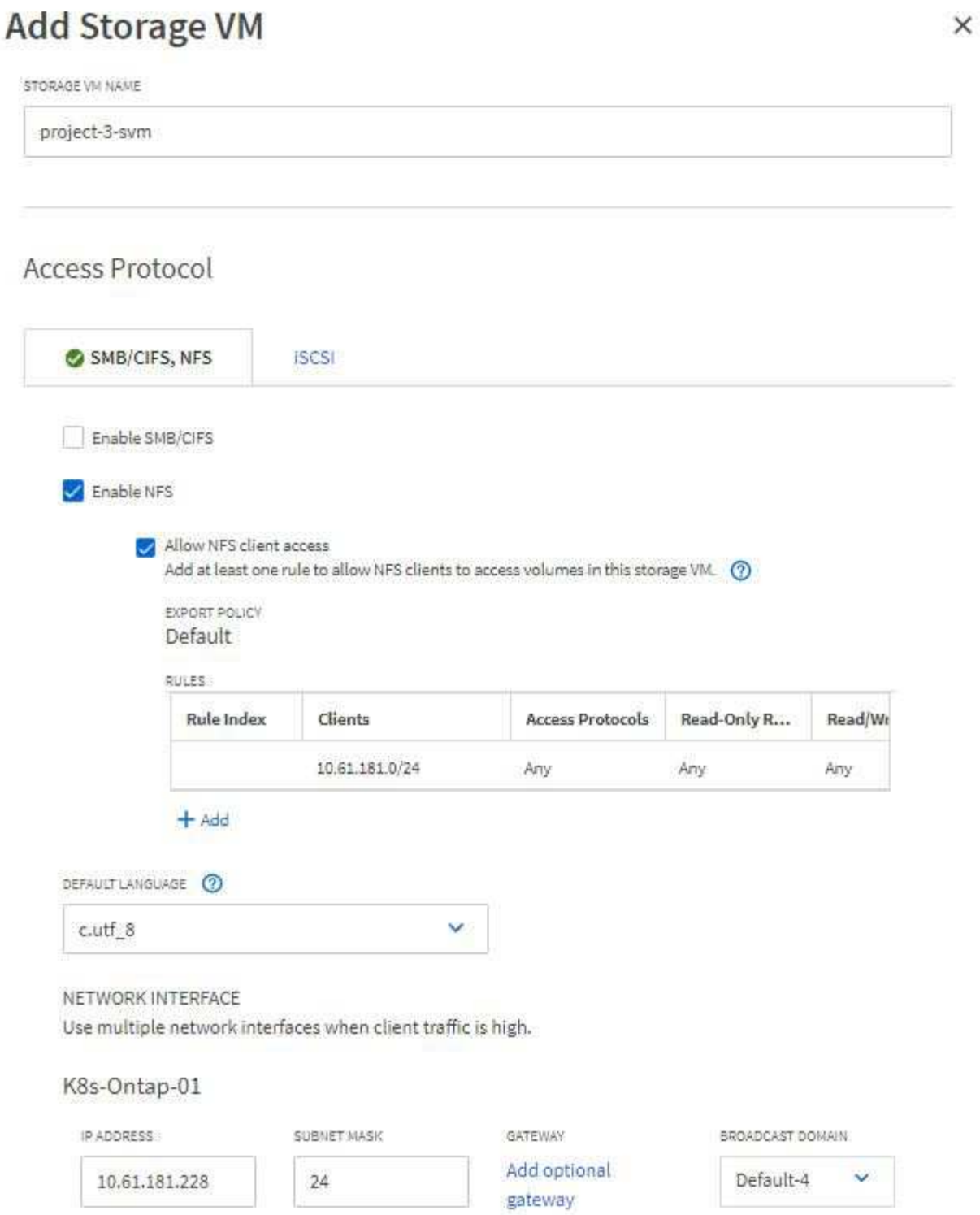
6. Vérifiez l'accès pour décrire les classes de stockage.
7. Valider l'accès de l'utilisateur pour modifier les classes de stockage.

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

Mise à l'échelle : ajout de projets supplémentaires

Dans une configuration multilocataire, l'ajout de nouveaux projets avec des ressources de stockage nécessite une configuration supplémentaire pour garantir que la multilocation n'est pas violée. Pour ajouter davantage de projets dans un cluster multilocataire, procédez comme suit :

1. Connectez-vous au cluster NetApp ONTAP en tant qu'administrateur de stockage.
2. Accéder à `Storage → Storage VMs` et cliquez `Add` . Créez un nouveau SVM dédié au projet-3. Créez également un compte vsadmin pour gérer le SVM et ses ressources.



1. Connectez-vous au cluster Red Hat OpenShift en tant qu'administrateur de cluster.
2. Créer un nouveau projet.

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

3. Assurez-vous que le groupe d’utilisateurs pour le projet 3 est créé sur IdP et synchronisé avec le cluster OpenShift.

```
oc get groups
```

4. Créez le rôle de développeur pour le projet-3.

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

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

La définition de rôle fournie dans cette section n'est qu'un exemple. Le rôle du développeur doit être défini en fonction des besoins de l'utilisateur final.

1. Créez RoleBinding pour les développeurs dans le projet-3 en liant le rôle developer-project-3 au groupe correspondant (ocp-project-3) dans le projet-3.

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

2. Connectez-vous au cluster Red Hat OpenShift en tant qu'administrateur de stockage
3. Créez un backend Trident et mappez-le au SVM dédié au projet-3. NetApp recommande d'utiliser le compte vsadmin du SVM pour connecter le backend au SVM au lieu d'utiliser l'administrateur du cluster ONTAP .

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

Nous utilisons le pilote ontap-nas pour cet exemple. Utilisez le pilote approprié pour créer le backend en fonction du cas d'utilisation.

Nous supposons que Trident est installé dans le projet Trident.

1. Créez la classe de stockage pour le projet-3 et configurez-la pour utiliser les pools de stockage du backend dédié au projet-3.

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

2. Créez un ResourceQuota pour restreindre les ressources du projet 3 demandant du stockage à partir de classes de stockage dédiées à d'autres projets.

```
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. Corrigez les ResourceQuotas dans d'autres projets pour empêcher les ressources de ces projets d'accéder au stockage à partir de la classe de stockage dédiée au projet-3.

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

Gestion avancée des clusters pour Kubernetes

Gestion avancée des clusters pour Kubernetes : Red Hat OpenShift avec NetApp – Présentation

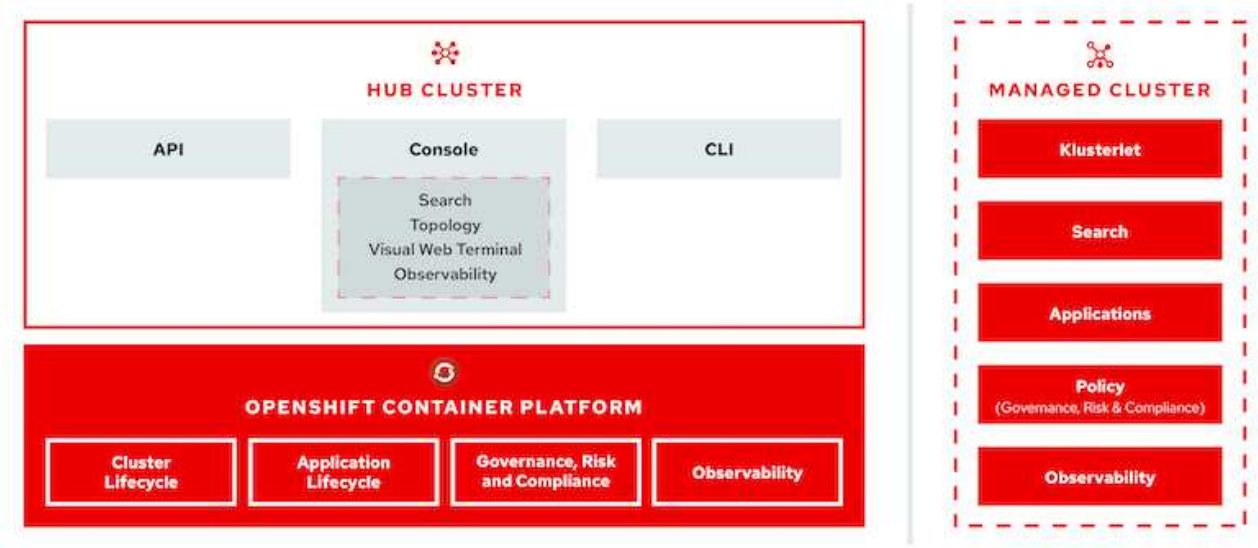
Lorsqu'une application conteneurisée passe du développement à la production, de nombreuses organisations ont besoin de plusieurs clusters Red Hat OpenShift pour prendre en charge les tests et le déploiement de cette application. Parallèlement à cela, les organisations hébergent généralement plusieurs applications ou charges de travail sur des clusters OpenShift. Par conséquent, chaque organisation finit par gérer un ensemble de clusters, et les administrateurs OpenShift doivent donc faire face au défi supplémentaire de gérer et de maintenir plusieurs clusters dans une gamme d'environnements qui s'étendent sur plusieurs centres de données sur site et clouds publics. Pour relever ces défis, Red Hat a introduit Advanced Cluster Management pour Kubernetes.

Red Hat Advanced Cluster Management pour Kubernetes vous permet d'effectuer les tâches suivantes :

1. Créez, importez et gérez plusieurs clusters dans des centres de données et des clouds publics
2. Déployez et gérez des applications ou des charges de travail sur plusieurs clusters à partir d'une seule console

3. Surveiller et analyser la santé et l'état des différentes ressources du cluster
4. Surveiller et appliquer la conformité de sécurité sur plusieurs clusters

Red Hat Advanced Cluster Management for Kubernetes est installé en tant que module complémentaire sur un cluster Red Hat OpenShift et utilise ce cluster comme contrôleur central pour toutes ses opérations. Ce cluster est connu sous le nom de cluster hub et expose un plan de gestion permettant aux utilisateurs de se connecter à Advanced Cluster Management. Tous les autres clusters OpenShift importés ou créés via la console Advanced Cluster Management sont gérés par le cluster hub et sont appelés clusters gérés. Il installe un agent appelé Klusterlet sur les clusters gérés pour les connecter au cluster hub et répondre aux demandes de différentes activités liées à la gestion du cycle de vie du cluster, à la gestion du cycle de vie des applications, à l'observabilité et à la conformité de sécurité.



Pour plus d'informations, consultez la documentation "ici" .

Déployer ACM pour Kubernetes

Déployer Advanced Cluster Management pour Kubernetes

Cette section couvre la gestion avancée des clusters pour Kubernetes sur Red Hat OpenShift avec NetApp.

Prérequis

1. Un cluster Red Hat OpenShift (supérieur à la version 4.5) pour le cluster hub
2. Clusters Red Hat OpenShift (supérieurs à la version 4.4.3) pour les clusters gérés
3. Accès administrateur du cluster au cluster Red Hat OpenShift
4. Un abonnement Red Hat pour Advanced Cluster Management pour Kubernetes

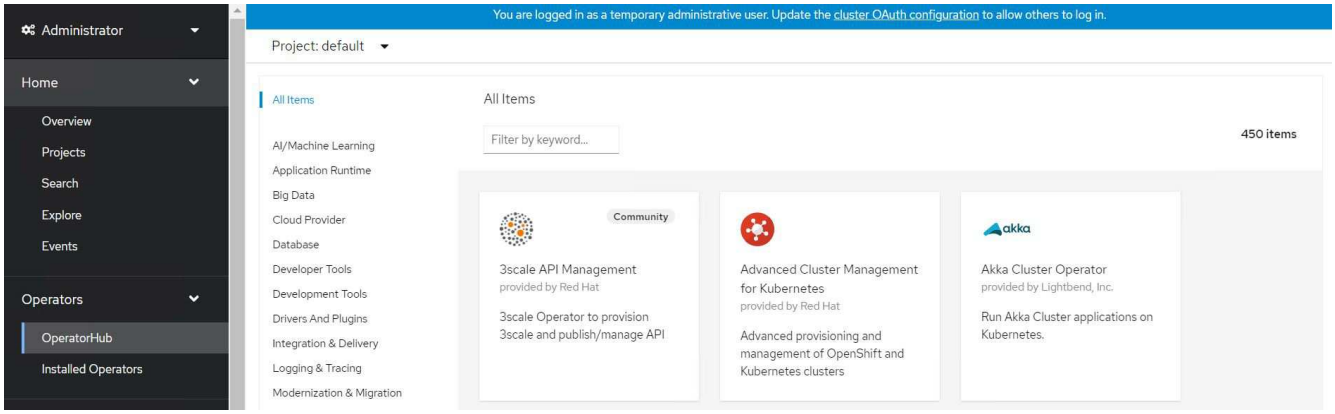
Advanced Cluster Management est un module complémentaire pour le cluster OpenShift. Il existe donc certaines exigences et restrictions sur les ressources matérielles en fonction des fonctionnalités utilisées sur le hub et les clusters gérés. Vous devez prendre en compte ces problèmes lors du dimensionnement des clusters. Voir la documentation "ici" pour plus de détails.

En option, si le cluster hub dispose de nœuds dédiés à l'hébergement des composants d'infrastructure et que vous souhaitez installer les ressources Advanced Cluster Management uniquement sur ces nœuds, vous devez ajouter des tolérances et des sélecteurs à ces nœuds en conséquence. Pour plus de détails, consultez la documentation "ici" .

Déployer Advanced Cluster Management pour Kubernetes

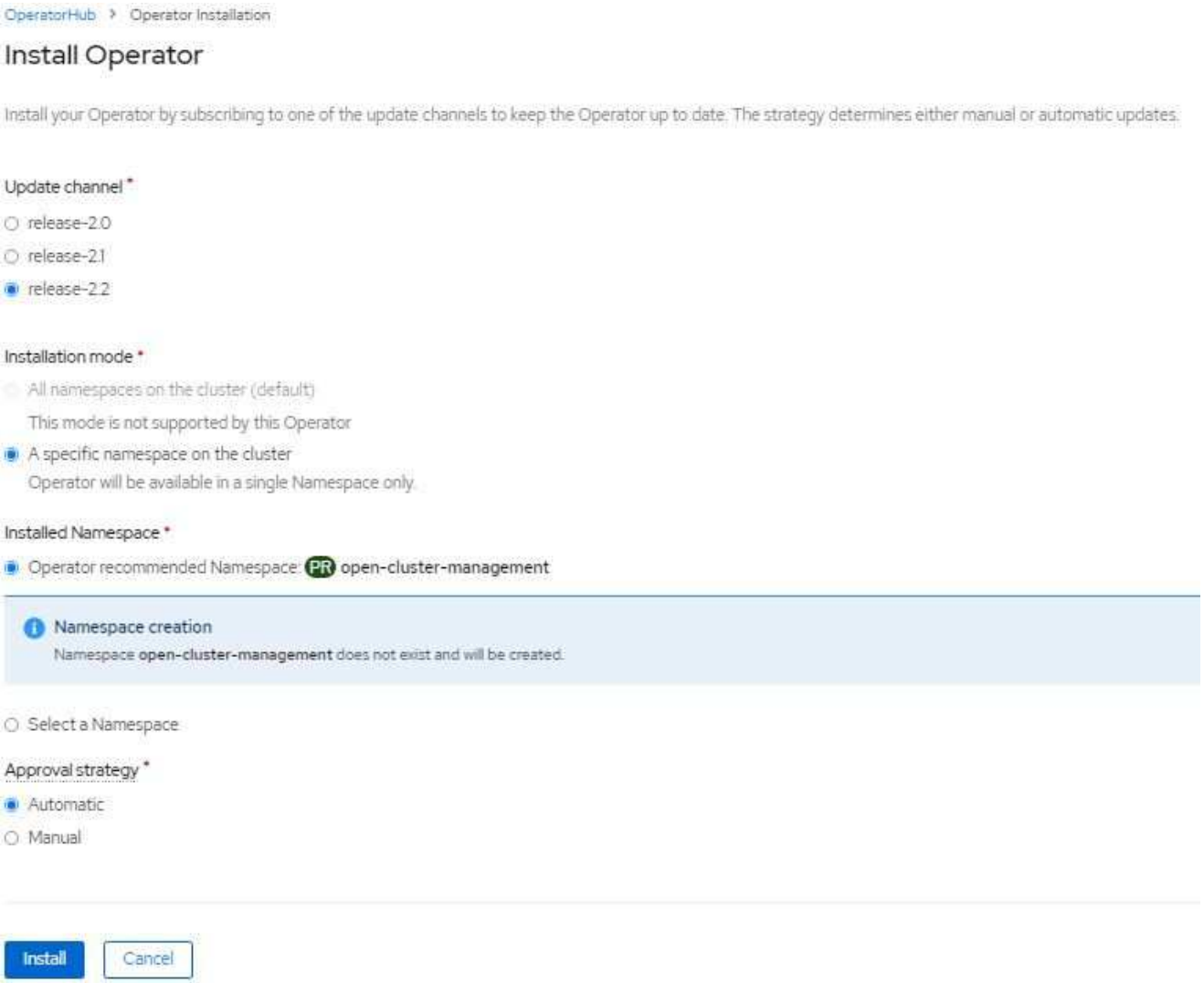
Pour installer Advanced Cluster Management pour Kubernetes sur un cluster OpenShift, procédez comme suit :

1. Choisissez un cluster OpenShift comme cluster hub et connectez-vous à celui-ci avec les privilèges d'administrateur de cluster.
2. Accédez à Opérateurs > Hub des opérateurs et recherchez Gestion avancée des clusters pour Kubernetes.

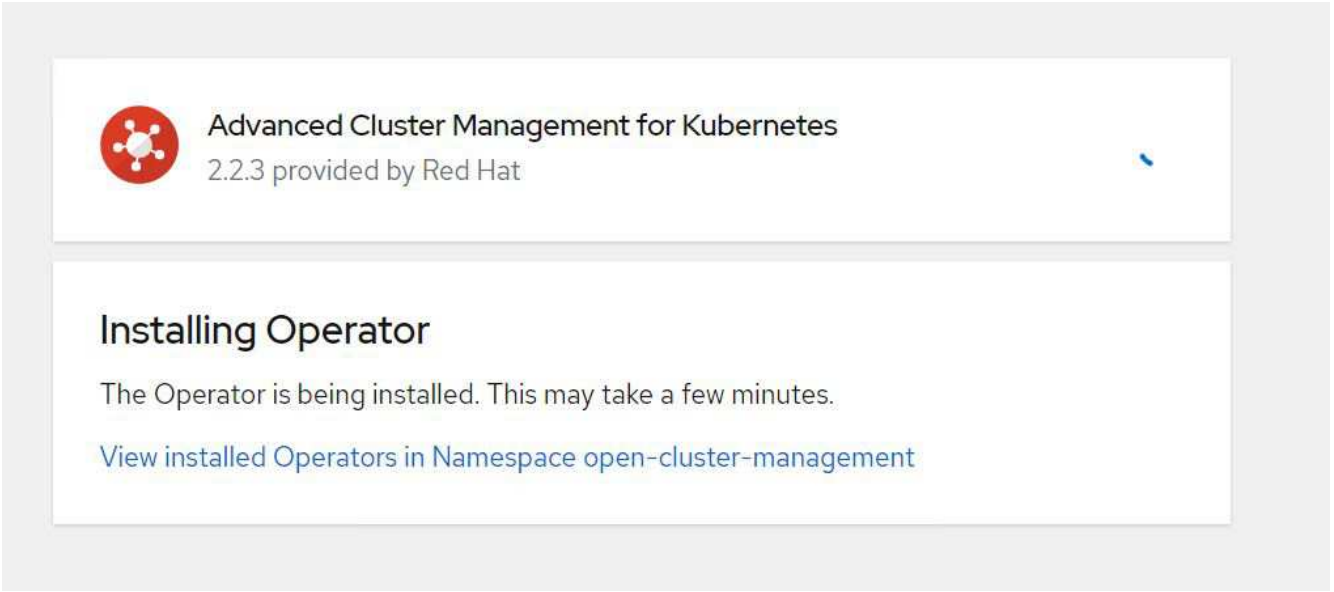


3. Sélectionnez Gestion avancée des clusters pour Kubernetes et cliquez sur Installer.

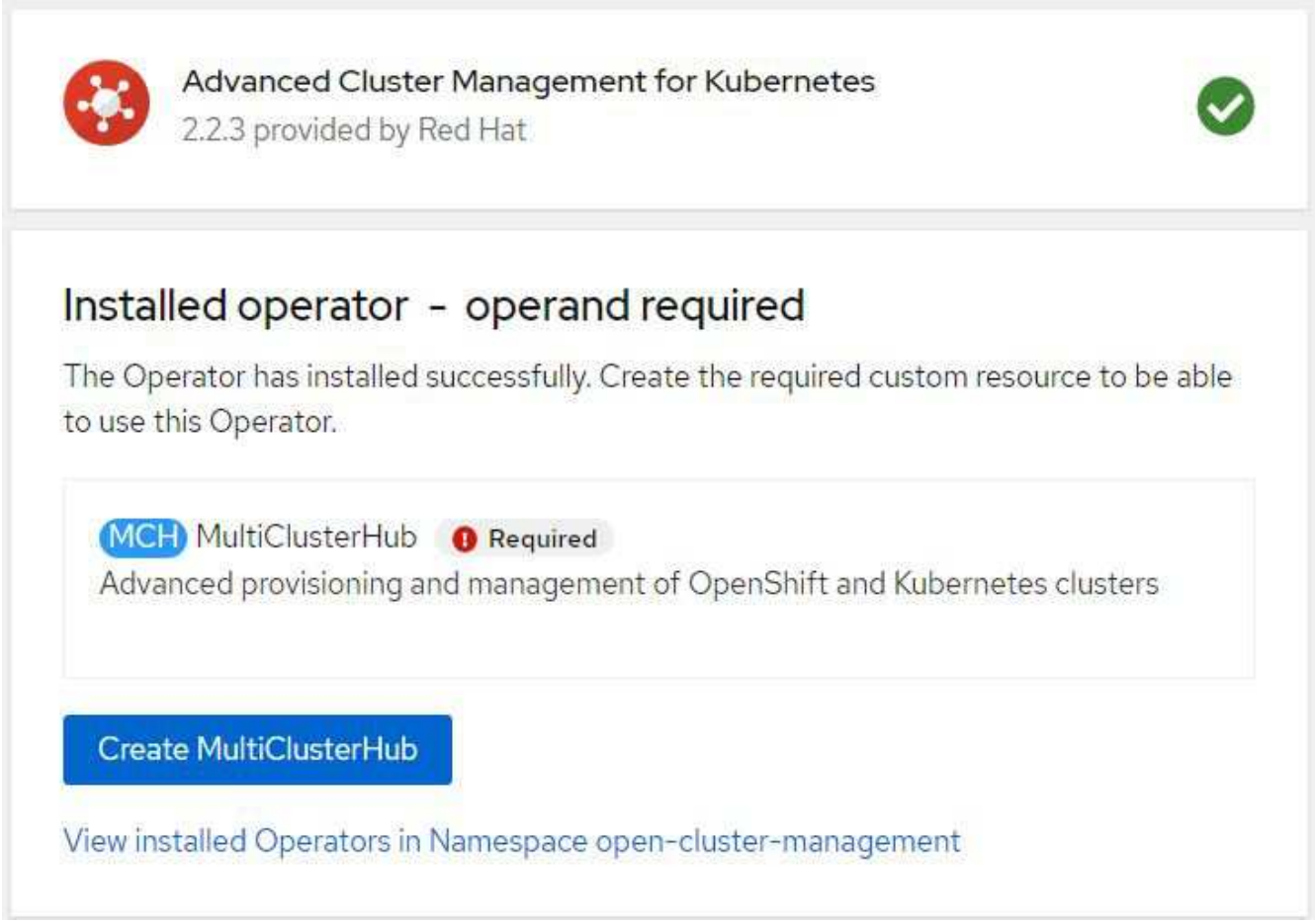
4. Sur l'écran Installer l'opérateur, fournissez les détails nécessaires (NetApp recommande de conserver les paramètres par défaut) et cliquez sur Installer.



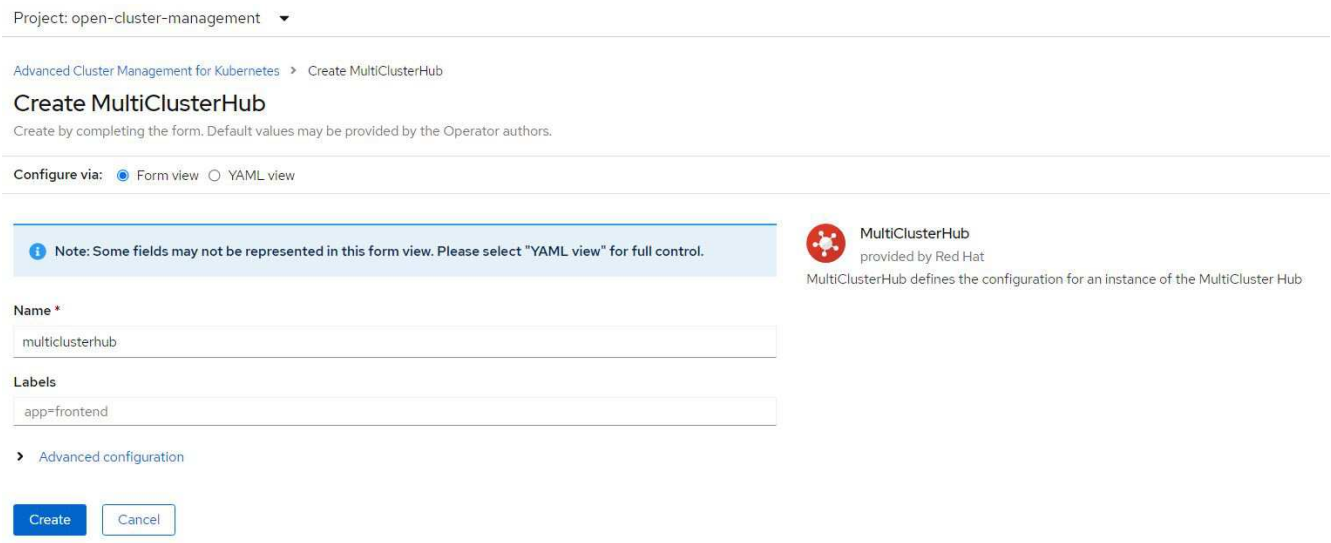
5. Attendez que l'installation de l'opérateur soit terminée.



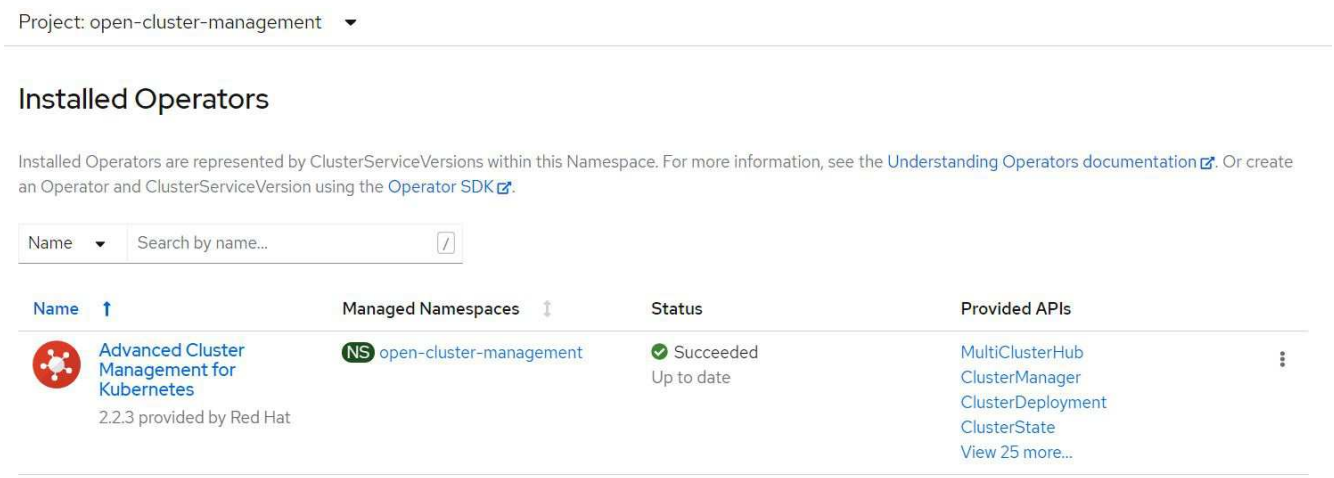
6. Une fois l'opérateur installé, cliquez sur Créer MultiClusterHub.



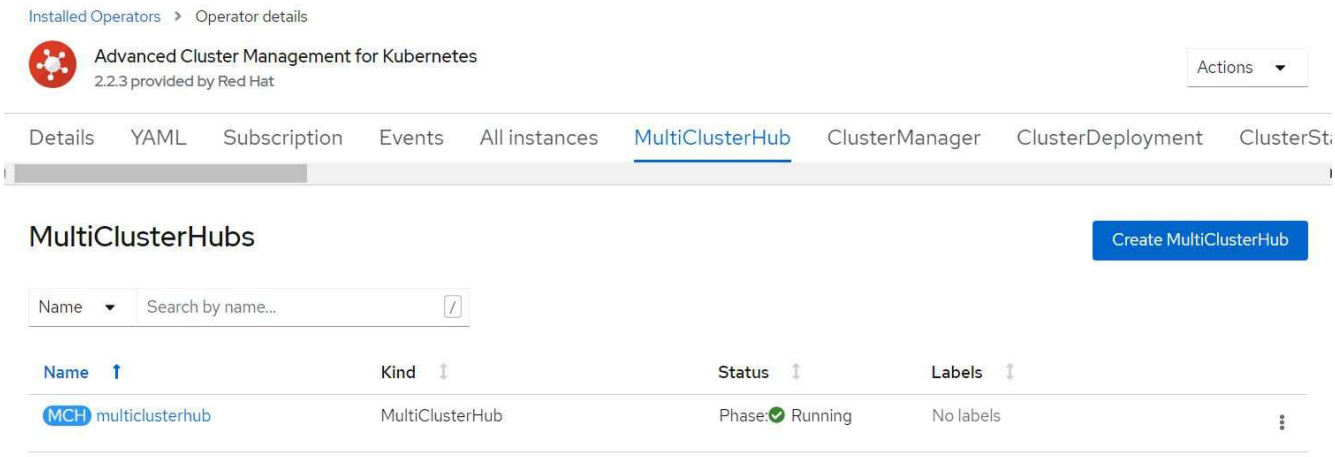
7. Sur l'écran Créer MultiClusterHub, cliquez sur Créer après avoir fourni les détails. Cela lance l'installation d'un hub multi-cluster.



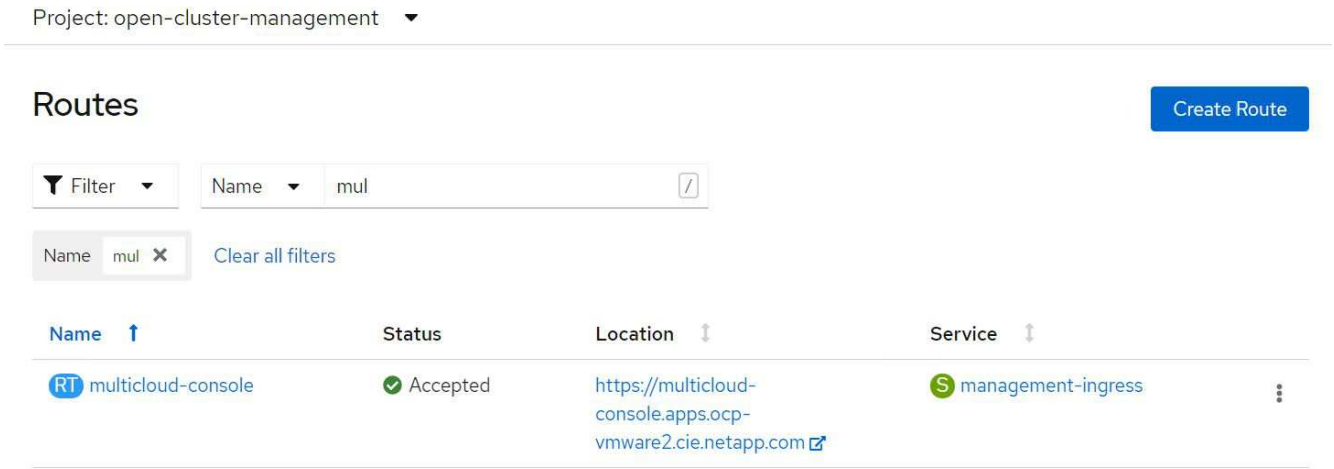
8. Une fois que tous les pods passent à l'état En cours d'exécution dans l'espace de noms open-cluster-management et que l'opérateur passe à l'état Réussi, Advanced Cluster Management pour Kubernetes est installé.



9. L'installation du hub prend un certain temps et, une fois celle-ci terminée, le hub MultiCluster passe à l'état d'exécution.



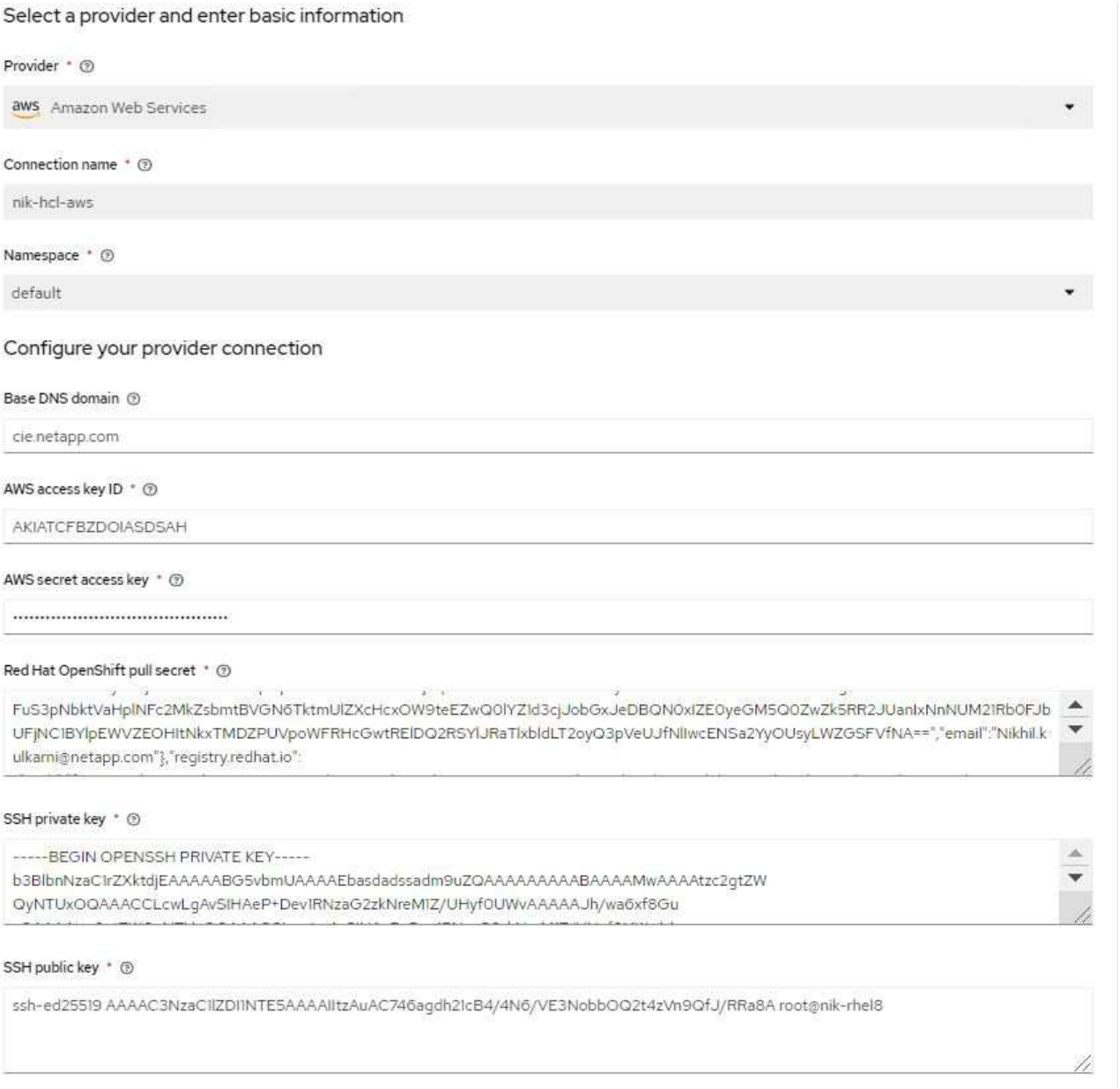
10. Il crée un itinéraire dans l'espace de noms open-cluster-management. Connectez-vous à l'URL de l'itinéraire pour accéder à la console Advanced Cluster Management.



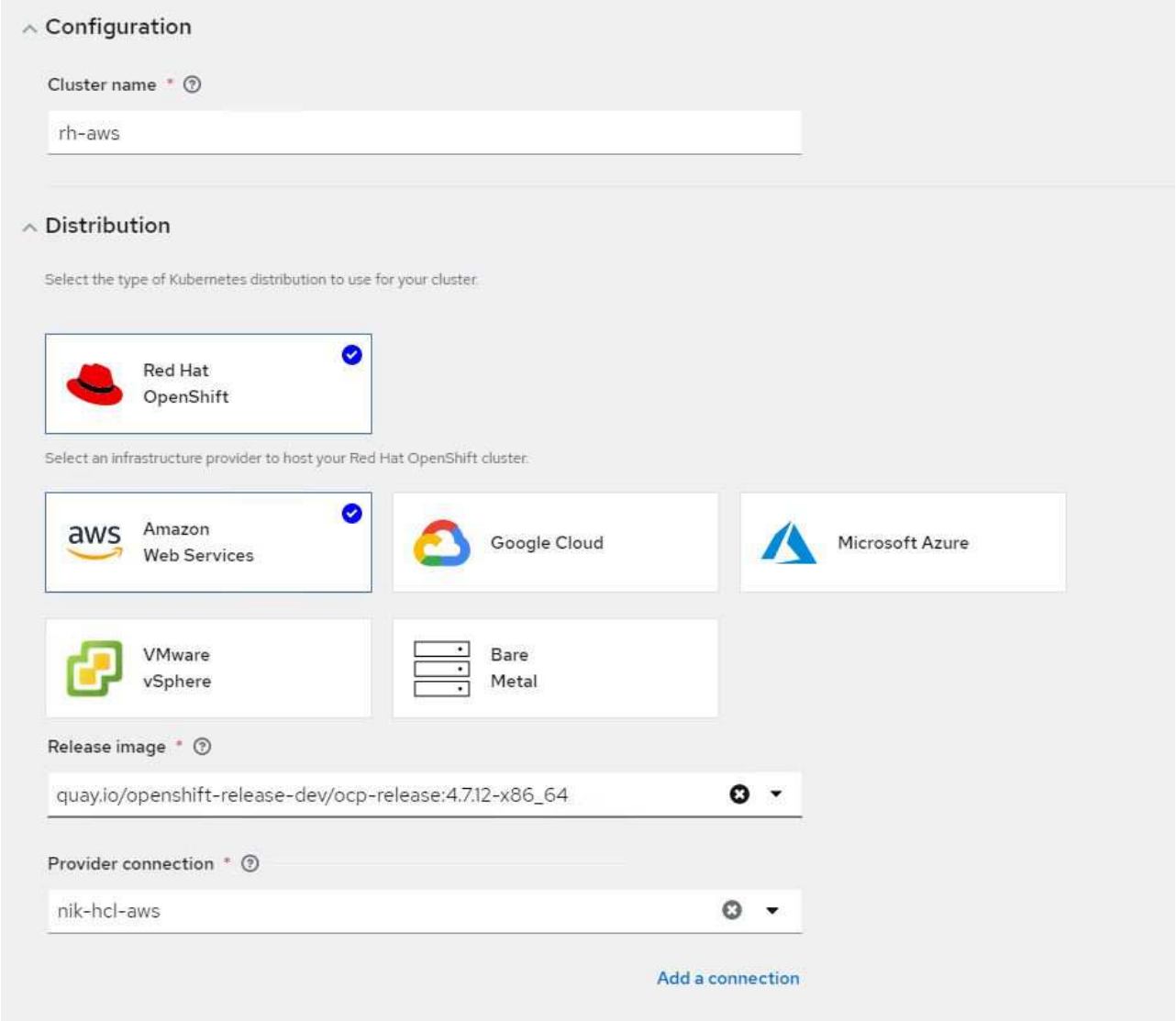
Gestion du cycle de vie des clusters

Pour gérer différents clusters OpenShift, vous pouvez les créer ou les importer dans Advanced Cluster Management.

1. Accédez d'abord à Automatiser les infrastructures > Clusters.
2. Pour créer un nouveau cluster OpenShift, procédez comme suit :
   a. Créer une connexion fournisseur : accédez à Connexions fournisseur et cliquez sur Ajouter une connexion, fournissez tous les détails correspondant au type de fournisseur sélectionné et cliquez sur Ajouter.

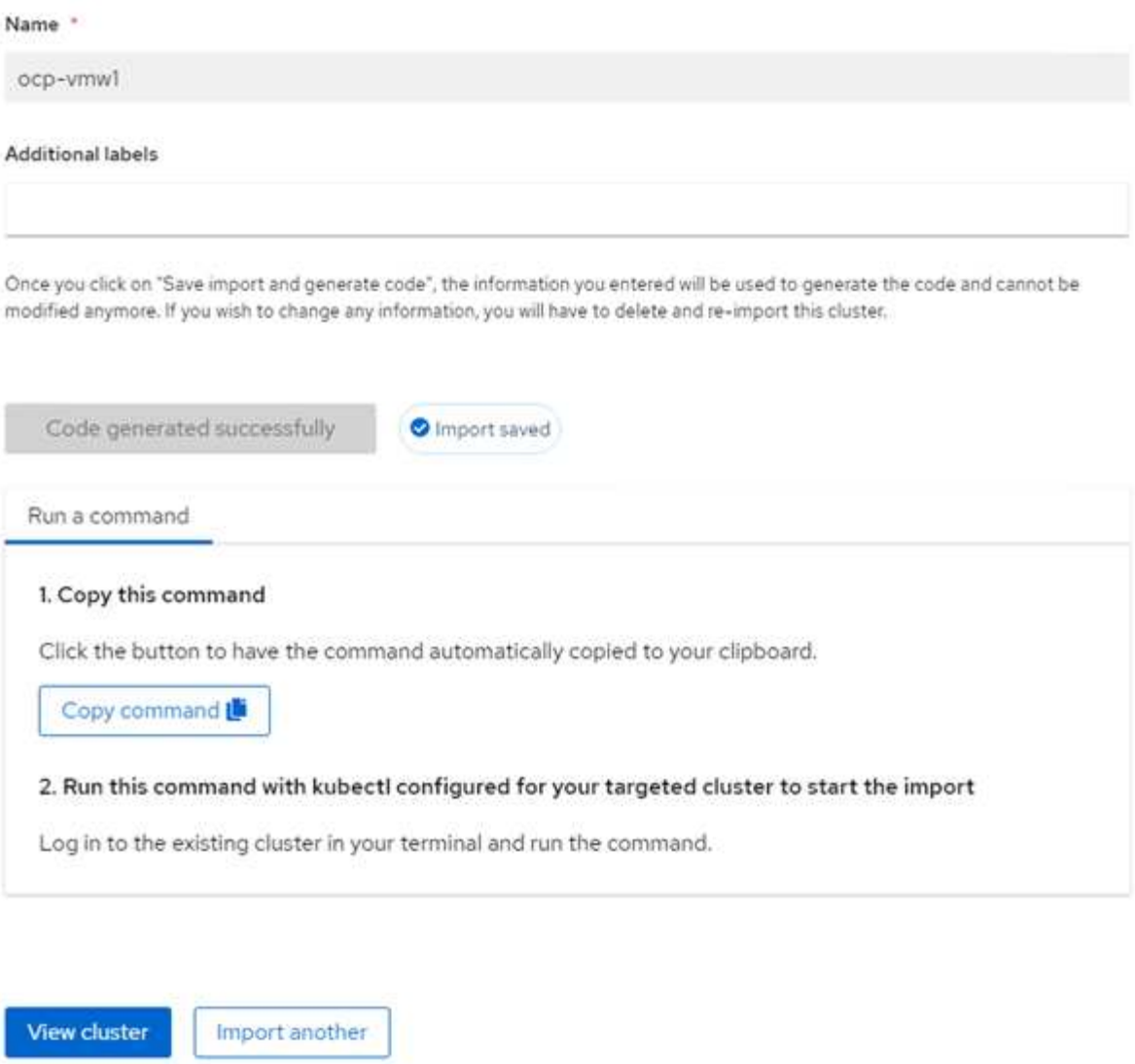


   b. Pour créer un nouveau cluster, accédez à Clusters et cliquez sur Ajouter un cluster > Créer un cluster. Fournissez les détails du cluster et du fournisseur correspondant et cliquez sur Créer.



c. Une fois le cluster créé, il apparaît dans la liste des clusters avec le statut Prêt.

3. Pour importer un cluster existant, procédez comme suit :
   a. Accédez à Clusters et cliquez sur Ajouter un cluster > Importer un cluster existant.
   b. Saisissez le nom du cluster et cliquez sur Enregistrer, Importer et Générer le code. Une commande permettant d'ajouter le cluster existant s'affiche.
   c. Cliquez sur Copier la commande et exécutez la commande sur le cluster à ajouter au cluster hub. Cela lance l'installation des agents nécessaires sur le cluster et, une fois ce processus terminé, le cluster apparaît dans la liste des clusters avec le statut Prêt.

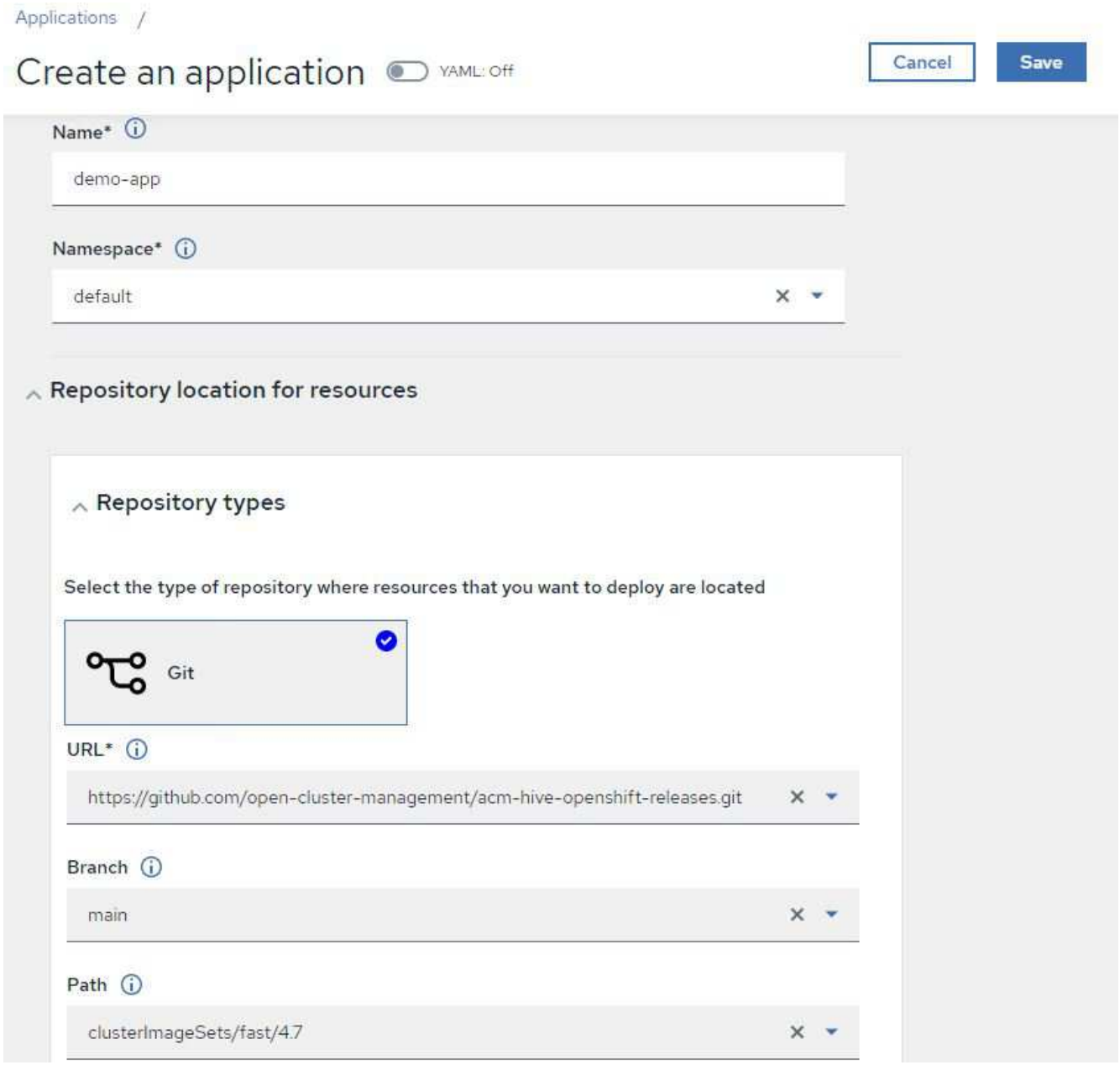


4. Après avoir créé et importé plusieurs clusters, vous pouvez les surveiller et les gérer à partir d'une seule console.

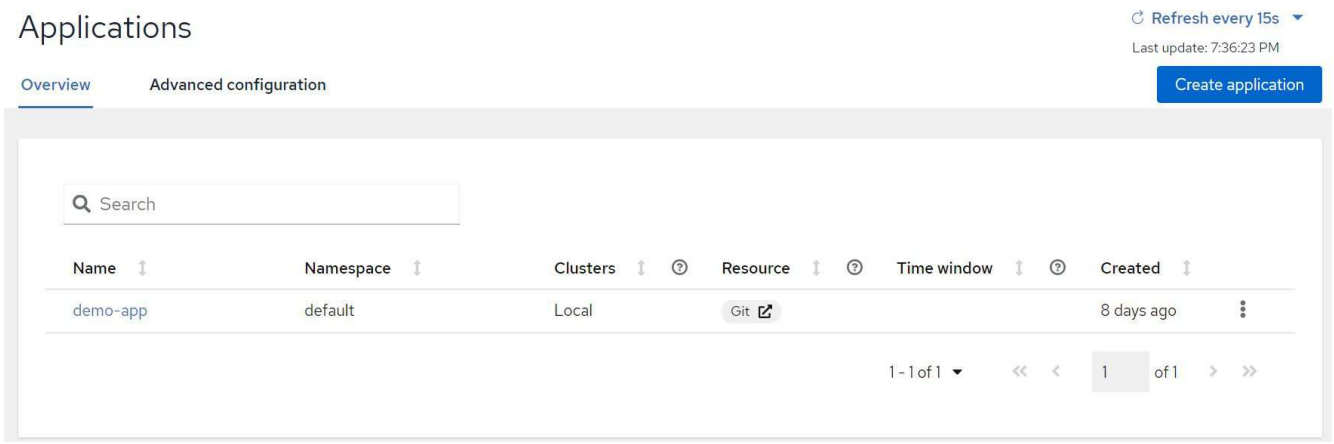
Gestion du cycle de vie des applications

Pour créer une application et la gérer sur un ensemble de clusters,

1. Accédez à Gérer les applications dans la barre latérale et cliquez sur Créer une application. Fournissez les détails de l'application que vous souhaitez créer et cliquez sur Enregistrer.



2. Une fois les composants de l'application installés, l'application apparaît dans la liste.

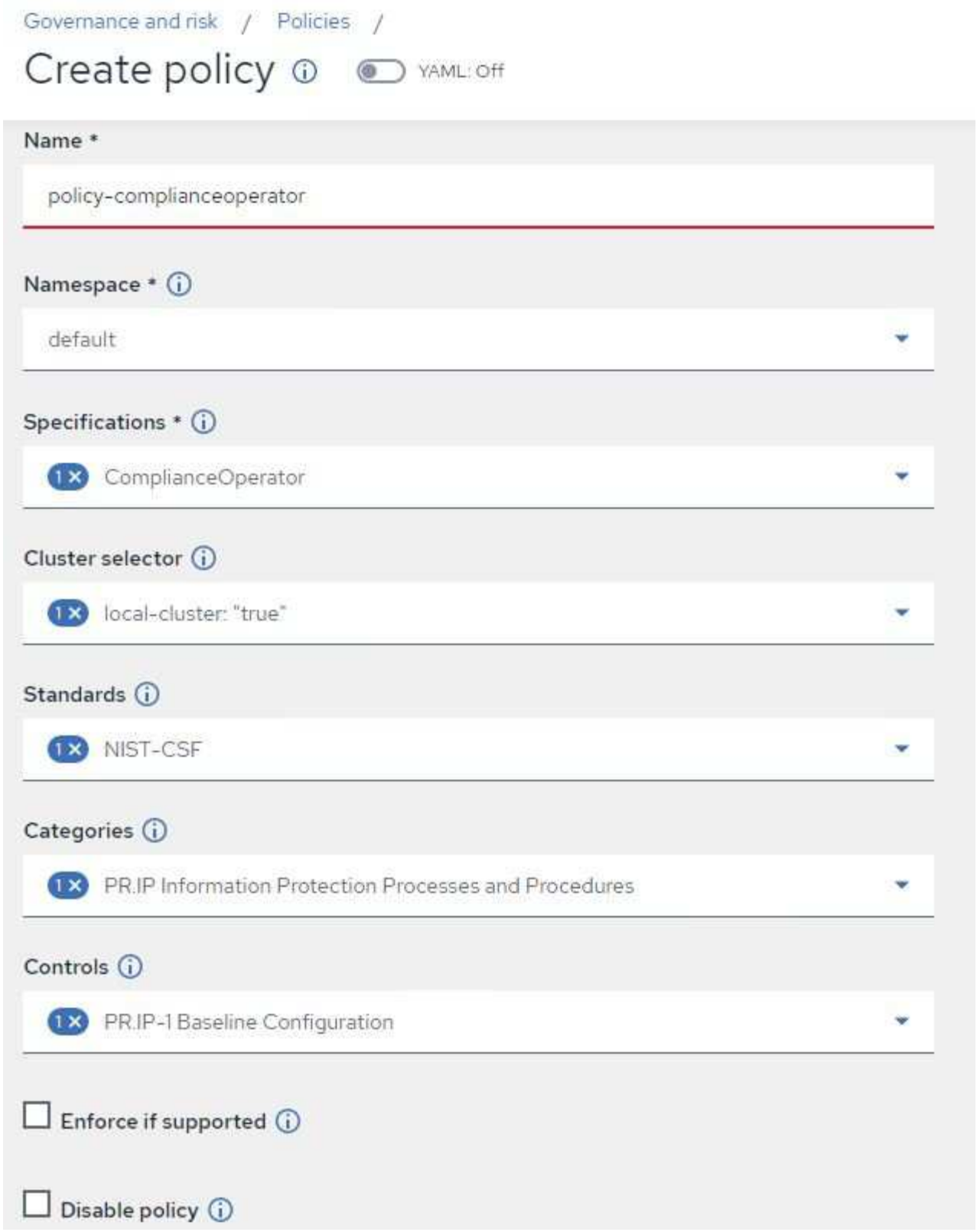


3. L'application peut désormais être surveillée et gérée depuis la console.

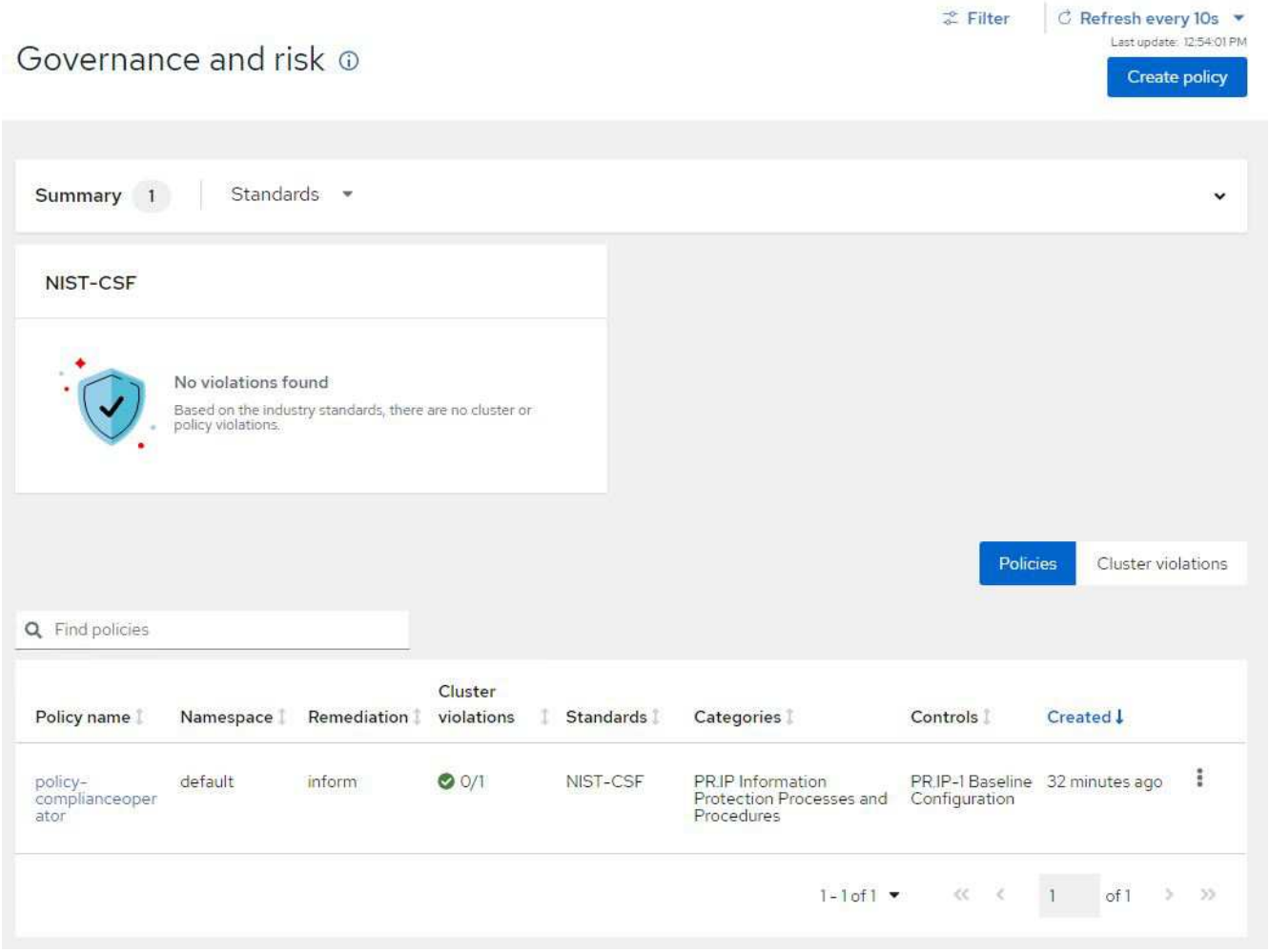
Gouvernance et risque

Cette fonctionnalité vous permet de définir les politiques de conformité pour différents clusters et de vous assurer que les clusters y adhèrent. Vous pouvez configurer les politiques pour informer ou corriger tout écart ou violation des règles.

1. Accédez à Gouvernance et risques depuis la barre latérale.
2. Pour créer des politiques de conformité, cliquez sur Créer une politique, entrez les détails des normes de politique et sélectionnez les clusters qui doivent adhérer à cette politique. Si vous souhaitez corriger automatiquement les violations de cette politique, cochez la case Appliquer si pris en charge et cliquez sur Créer.



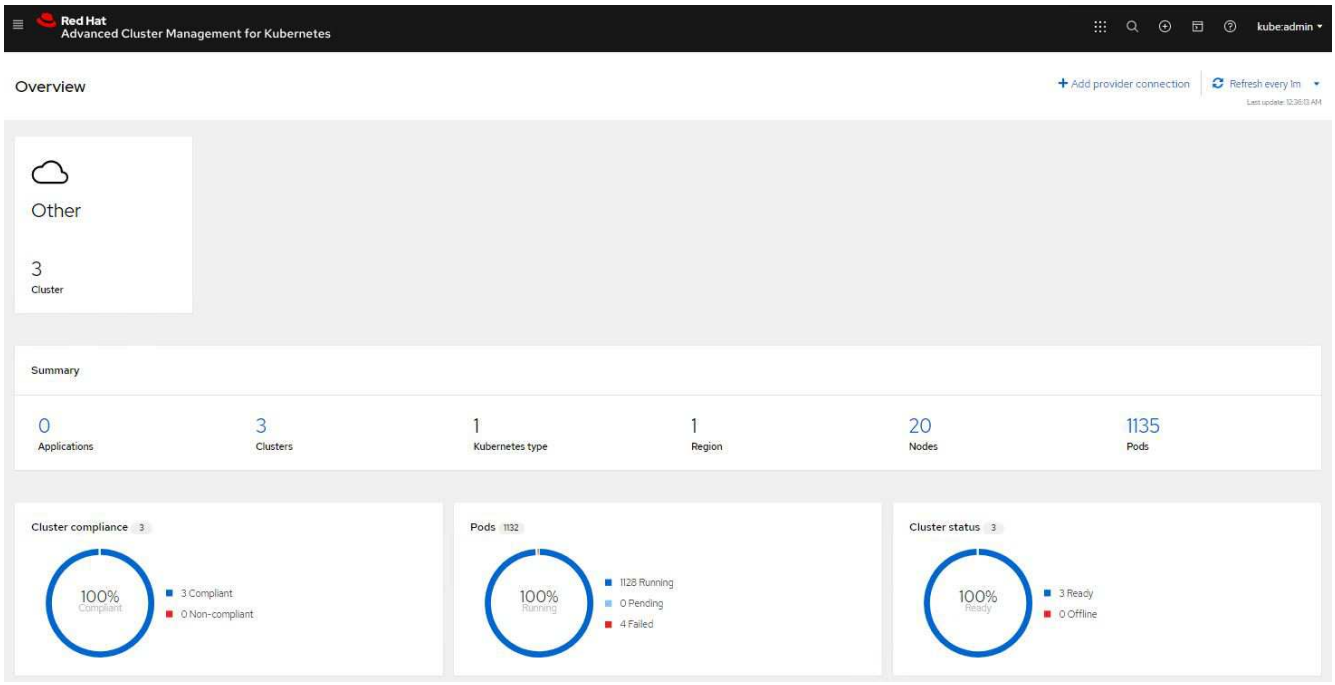
3. Une fois toutes les politiques requises configurées, toutes les violations de politique ou de cluster peuvent être surveillées et corrigées à partir de Advanced Cluster Management.



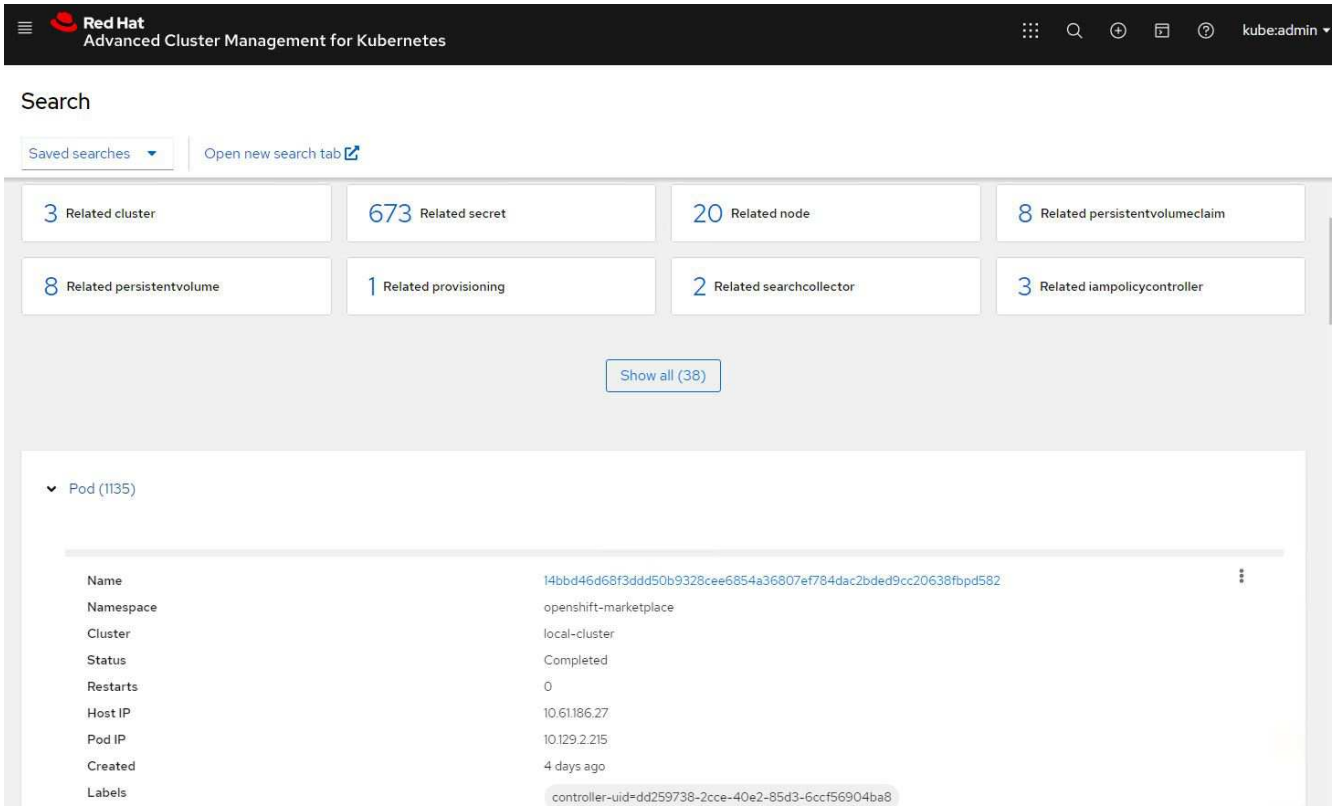
Observabilité

Advanced Cluster Management pour Kubernetes offre un moyen de surveiller les nœuds, les pods, les applications et les charges de travail sur tous les clusters.

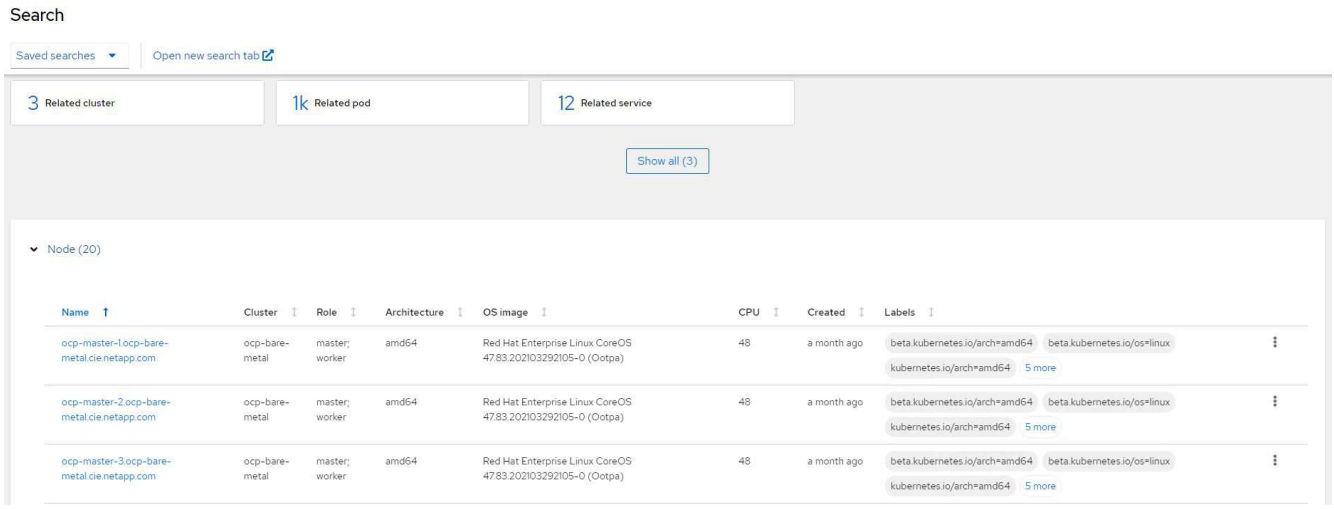
1. Accédez à Observer les environnements > Présentation.



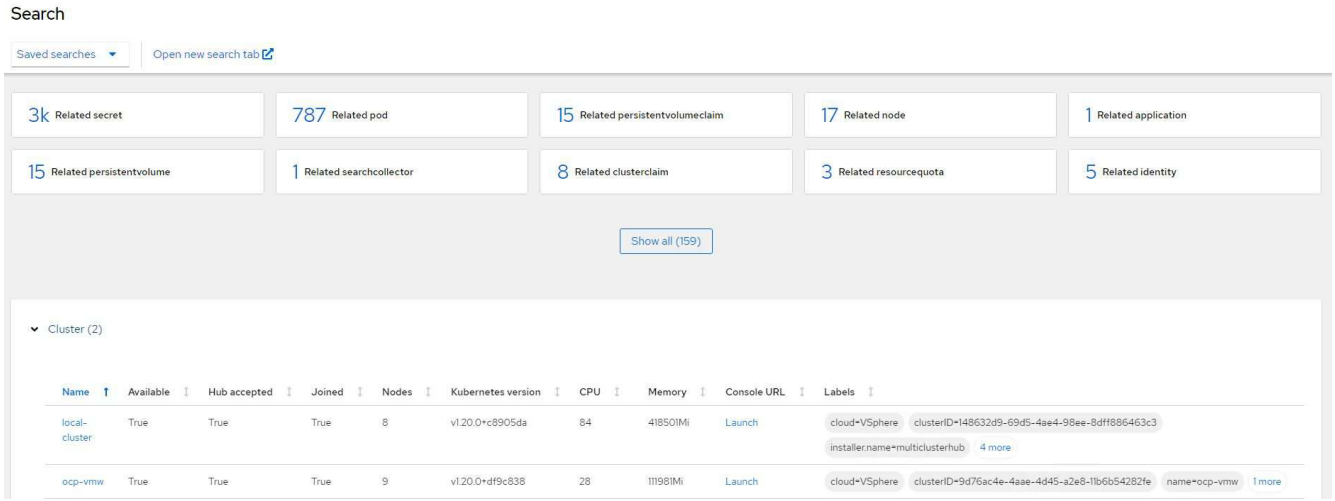
2. Tous les pods et charges de travail de tous les clusters sont surveillés et triés en fonction de divers filtres. Cliquez sur Pods pour afficher les données correspondantes.



3. Tous les nœuds des clusters sont surveillés et analysés en fonction de divers points de données. Cliquez sur Nœuds pour obtenir plus d'informations sur les détails correspondants.

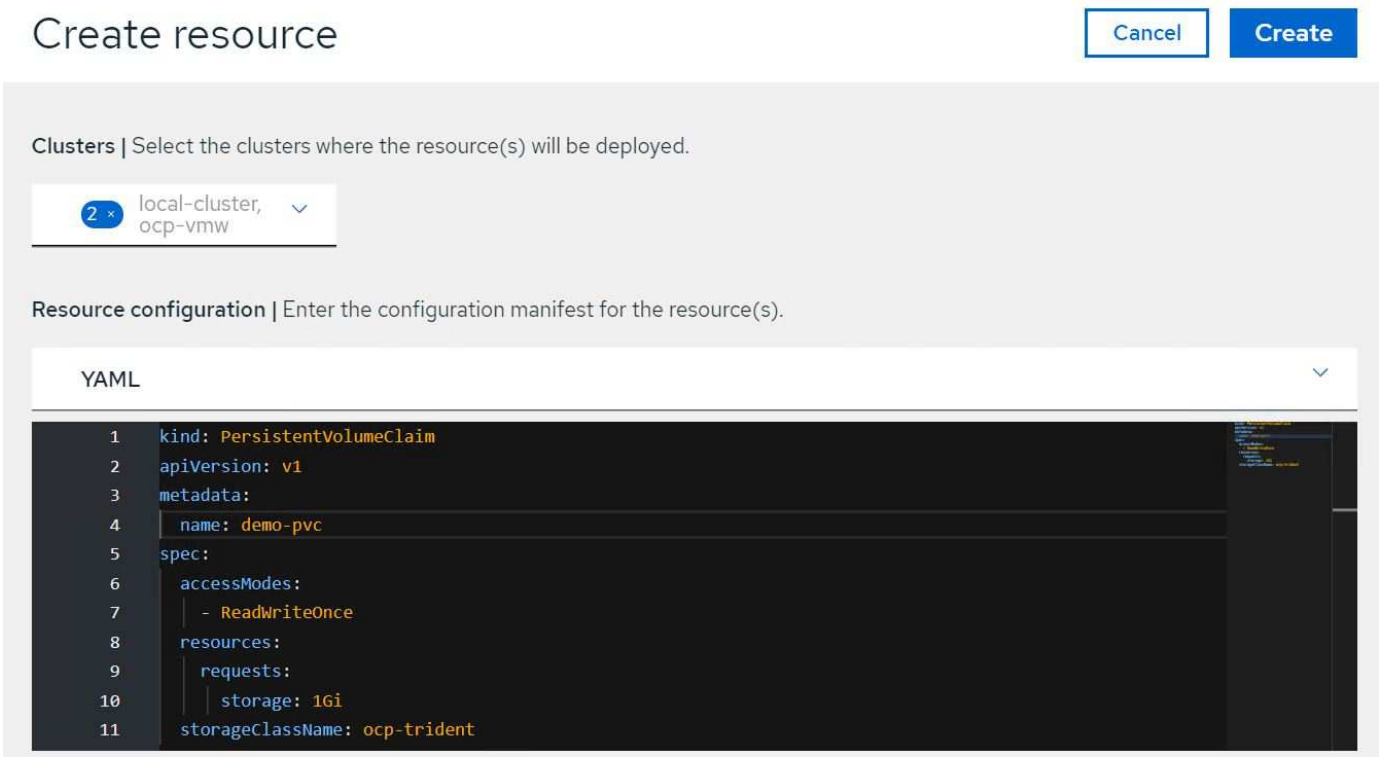


4. Tous les clusters sont surveillés et organisés en fonction de différentes ressources et paramètres de cluster. Cliquez sur Clusters pour afficher les détails du cluster.



Créer des ressources sur plusieurs clusters

Advanced Cluster Management pour Kubernetes permet aux utilisateurs de créer des ressources sur un ou plusieurs clusters gérés simultanément à partir de la console. Par exemple, si vous avez des clusters OpenShift sur différents sites soutenus par différents clusters NetApp ONTAP et que vous souhaitez provisionner des PVC sur les deux sites, vous pouvez cliquer sur le signe (+) dans la barre supérieure. Sélectionnez ensuite les clusters sur lesquels vous souhaitez créer le PVC, collez le YAML de la ressource et cliquez sur Créer.



Protection des données pour les applications conteneurisées et les machines virtuelles à l'aide de Trident Protect

Cette solution montre comment utiliser Trident Protect pour effectuer des opérations de protection des données pour les conteneurs et les machines virtuelles.

1. Pour plus de détails sur la création d'instantanés et de sauvegardes et leur restauration pour les applications de conteneur dans la plateforme OpenShift Container, reportez-vous à"ici" .
2. Pour plus de détails sur la création et la restauration à partir d'une sauvegarde pour les machines virtuelles dans OpenShift Virtualization déployées sur la plateforme OpenShift Container, reportez-vous à"ici" .

Protection des données pour les applications conteneurisées et les machines virtuelles à l'aide d'outils tiers

Cette solution montre comment utiliser Velero qui est intégré à l'opérateur OADP dans la plateforme Red Hat OpenShift Container pour effectuer des opérations de protection des données pour les conteneurs et les machines virtuelles.

1. Pour plus de détails sur la création et la restauration à partir d'une sauvegarde pour les applications de conteneur dans la plateforme OpenShift Container, reportez-vous à"ici" .
2. Pour plus de détails sur la création et la restauration à partir d'une sauvegarde pour les machines virtuelles dans OpenShift Virtualization déployées sur la plateforme OpenShift Container, reportez-vous à"ici" .

Ressources supplémentaires pour en savoir plus sur l'intégration de Red Hat OpenShift Virtualization avec le stockage NetApp

Accédez à des ressources supplémentaires qui offrent plus d'informations sur la prise en charge du déploiement, de la gestion et de l'optimisation de Red Hat OpenShift Virtualization avec ONTAP sur différentes plates-formes et technologies.

- Documentation NetApp

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

- Documentation Trident

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

- Documentation Red Hat OpenShift

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

- Documentation de la plateforme Red Hat OpenStack

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

- Documentation sur la virtualisation Red Hat

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

- Documentation VMware vSphere

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

Informations sur le copyright

Informations sur les marques commerciales