OpenShift Container Platform 4.12

Virtualisation

Installation, utilisation et notes de version d'OpenShift Virtualization

Last Updated: 2023-05-04

OpenShift Container Platform 4.12 Virtualisation

Installation, utilisation et notes de version d'OpenShift Virtualization

Notice légale

Résumé

Ce document fournit des informations sur l'utilisation d'OpenShift Virtualization dans OpenShift Container Platform.

Table des matières

CHAPITRE 1. À PROPOS DE LA VIRTUALISATION OPENSHIFT

Découvrez les capacités d'OpenShift Virtualization et l'étendue de la prise en charge.

1.1. CE QUE VOUS POUVEZ FAIRE AVEC OPENSHIFT VIRTUALIZATION

OpenShift Virtualization est un module complémentaire à OpenShift Container Platform qui vous permet d'exécuter et de gérer des charges de travail de machines virtuelles en même temps que des charges de travail de conteneurs.

OpenShift Virtualization ajoute de nouveaux objets dans votre cluster OpenShift Container Platform en utilisant les ressources personnalisées de Kubernetes pour activer les tâches de virtualisation. Ces tâches comprennent :

- Création et gestion de machines virtuelles Linux et Windows
- Connexion aux machines virtuelles à l'aide d'une variété de consoles et d'outils CLI
- Importation et clonage de machines virtuelles existantes
- Gestion des contrôleurs d'interface réseau et des disques de stockage attachés aux machines virtuelles
- Migration en direct des machines virtuelles entre les nœuds

Une console web améliorée fournit un portail graphique pour gérer ces ressources virtualisées ainsi que les conteneurs et l'infrastructure du cluster OpenShift Container Platform.

OpenShift Virtualization est conçu et testé pour fonctionner correctement avec les fonctionnalités de Red Hat OpenShift Data Foundation.

Vous pouvez utiliser OpenShift Virtualization avec OVN-Kubernetes, OpenShift SDN, ou l'un des autres plugins réseau certifiés listés dans Certified OpenShift CNI Plug-ins.

Vous pouvez vérifier la conformité de votre cluster OpenShift Virtualization en installant l'opérateur de conformité et en lançant un scan avec les profils **ocp4-moderate** et **ocp4-moderate-node**. L'opérateur de conformité utilise OpenSCAP, un outil certifié par le NIST , pour analyser et appliquer les politiques de sécurité.

IMPORTANT

L'intégration d'OpenShift Virtualization avec l'opérateur de conformité est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

1.1.1. Version de cluster prise en charge par OpenShift Virtualization

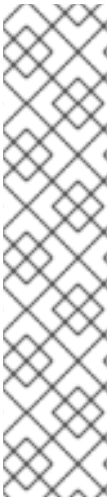
OpenShift Virtualization 4.12 est pris en charge pour une utilisation sur les clusters OpenShift Container Platform 4.12. Pour utiliser la dernière version z-stream d'OpenShift Virtualization, vous devez d'abord passer à la dernière version d'OpenShift Container Platform.

1.2. DIFFÉRENCES ENTRE OPENSHIFT ET UN SEUL NŒUD

Vous pouvez installer OpenShift Virtualization sur un cluster à nœud unique.

Lors du provisionnement d'un cluster OpenShift à nœud unique avec l'installateur assisté, le stockage persistant préconfiguré est déployé automatiquement.

- Dans OpenShift Virtualization 4.10 et 4.11, le HostPath Provisioner (HPP) est automatiquement installé.
- Dans OpenShift Virtualization 4.12, l'opérateur OpenShift Data Foundation Logical Volume Manager est la solution de stockage prête à l'emploi. Vous pouvez également procéder à un déploiement manuel à l'aide du HPP.



NOTE

OpenShift à nœud unique ne prend pas en charge la haute disponibilité. Soyez conscient des différences de fonctionnalité suivantes par rapport à un cluster à nœuds multiples :

- Les budgets d'interruption de la production ne sont pas pris en charge.
- La migration en direct n'est pas prise en charge.
- En raison de différences dans le comportement du stockage, certains modèles de machines virtuelles sont incompatibles avec OpenShift à nœud unique. Pour assurer la compatibilité, les modèles ou les machines virtuelles qui utilisent des volumes de données ou des profils de stockage ne doivent pas avoir la stratégie d'éviction définie.

1.3. RESSOURCES SUPPLÉMENTAIRES

- À propos d'OpenShift à nœud unique
- Installateur assisté
- Hostpath Provisioner (HPP)
- OpenShift Container Platform Data Foundation Logical Volume Manager Opérateur
- Budgets pour les perturbations des gousses
- Migration en direct
- Stratégie d'expulsion

CHAPITRE 2. ARCHITECTURE DE VIRTUALISATION OPENSHIFT

Découvrez l'architecture de virtualisation d'OpenShift.

2.1. COMMENT FONCTIONNE L'ARCHITECTURE DE VIRTUALISATION OPENSHIFT

Après avoir installé OpenShift Virtualization, l'Operator Lifecycle Manager (OLM) déploie des pods d'opérateur pour chaque composant d'OpenShift Virtualization :

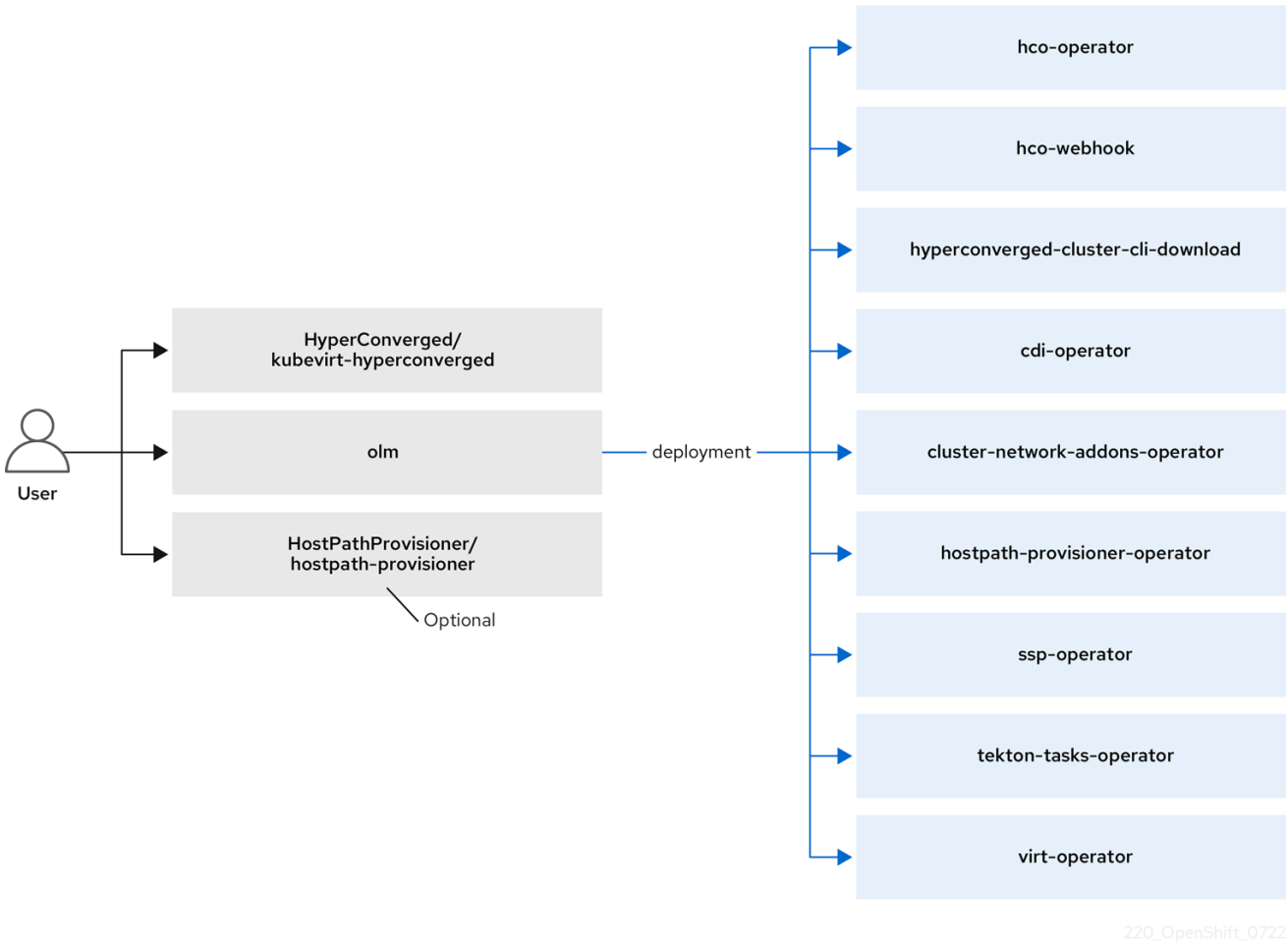
- Calculer : **virt-operator**
- Stockage : **cdi-operator**
- Réseau : **cluster-network-addons-operator**
- Échelle : **ssp-operator**
- Templating : **tekton-tasks-operator**

OLM déploie également le pod **hyperconverged-cluster-operator**, qui est responsable du déploiement, de la configuration et du cycle de vie des autres composants, ainsi que plusieurs pods d'aide : **hco-webhook**, et **hyperconverged-cluster-cli-download**.

Une fois que tous les pods opérateurs ont été déployés avec succès, vous devez créer la ressource personnalisée (CR) **HyperConverged**. Les configurations définies dans la CR **HyperConverged** servent de source unique de vérité et de point d'entrée pour OpenShift Virtualization, et guident le comportement des CR.

Le CR **HyperConverged** crée des CR correspondants pour les opérateurs de tous les autres composants dans sa boucle de réconciliation. Chaque opérateur crée ensuite des ressources telles que des ensembles de démons, des cartes de configuration et des composants supplémentaires pour le plan de contrôle d'OpenShift Virtualization. Par exemple, lorsque le **hco-operator** crée le CR **KubeVirt**, le **virt-operator** le réconcilie et crée des ressources supplémentaires telles que **virt-controller**, **virt-handler** et **virt-api**.

L'OLM déploie le site **hostpath-provisioner-operator**, mais il n'est pas fonctionnel tant que vous n'avez pas créé un CR **hostpath provisioner** (HPP).



Ressources supplémentaires

- Configuration de la CR hyperconvergée
- Commandes du client Virtctl

2.2. À PROPOS DE L'OPÉRATEUR HCO

Le site **hco-operator** (HCO) fournit un point d'entrée unique pour déployer et gérer OpenShift Virtualization et plusieurs opérateurs d'aide avec des valeurs par défaut basées sur l'opinion. Il crée également des ressources personnalisées (CR) pour ces opérateurs.

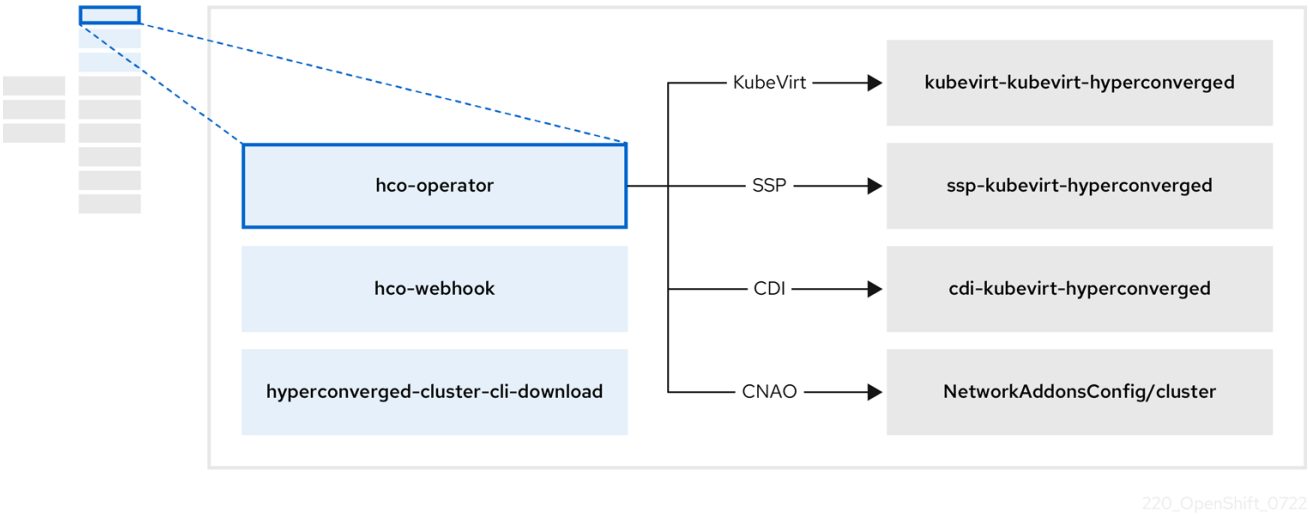


Tableau 2.1. composants de l'opérateur hco

Component	Description
deployment/hco-webhook	Valide le contenu de la ressource personnalisée **HyperConverged**.
deployment/hyperconverged-cluster-cli-download	Fournit les binaires de l'outil **virtctl** au cluster afin que vous puissiez les télécharger directement depuis le cluster.
KubeVirt/kubevirt-kubevirt-hyperconverged	Contient tous les opérateurs, CR et objets nécessaires à OpenShift Virtualization.
SSP/ssp-kubevirt-hyperconverged	Un CR du PAS. Il est automatiquement créé par le HCO.
CDI/cdi-kubevirt-hyperconverged	UN CDI CR. Celui-ci est automatiquement créé par le HCO.
NetworkAddonsConfig/cluster	Un CR qui donne des instructions et est géré par le **cluster-network-addons-operator**.

2.3. A PROPOS DU CDI-OPÉRATEUR

Le site **cdi-operator** gère l'importateur de données conteneurisées (CDI) et ses ressources connexes, qui importent une image de machine virtuelle (VM) dans une revendication de volume persistant (PVC) à l'aide d'un volume de données.

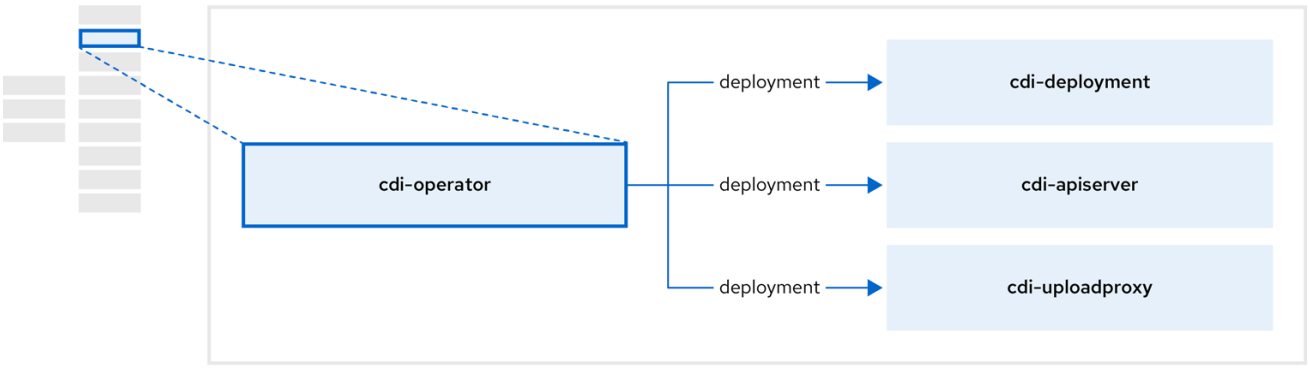


Tableau 2.2. composants du cdi-opérateur

Component	Description
deployment/cdi-apiserver	Gère l'autorisation de télécharger des disques VM dans des PVC en émettant des jetons de téléchargement sécurisés.
deployment/cdi-uploadproxy	Dirige le trafic de téléchargement sur disque externe vers le pod de serveur de téléchargement approprié afin qu'il puisse être écrit sur le bon PVC. Nécessite un jeton de téléchargement valide.
pod/cdi-importer	Pod d'aide qui importe une image de machine virtuelle dans un PVC lors de la création d'un volume de données.

2.4. À PROPOS DE L'OPÉRATEUR CLUSTER-NETWORK-ADDONS-OPERATOR

Le site **cluster-network-addons-operator** déploie des composants réseau sur une grappe et gère les ressources correspondantes pour une fonctionnalité réseau étendue.

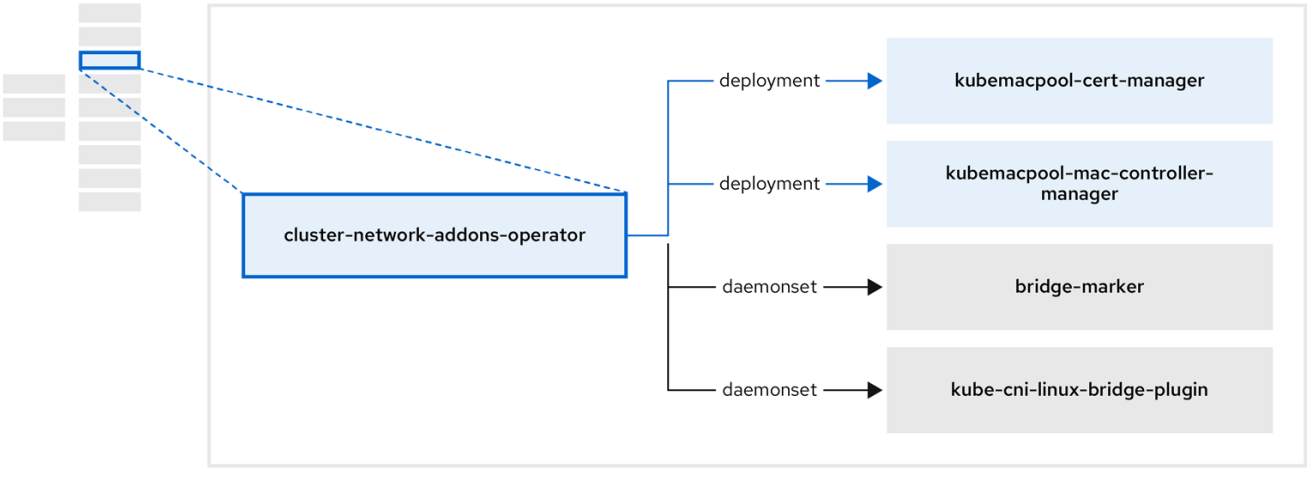


Tableau 2.3. composants de l'opérateur cluster-network-addons

Component	Description
deployment/kubemacpool-cert-manager	Gère les certificats TLS des webhooks de Kubemacpool.
deployment/kubemacpool-mac-controller-manager	Fournit un service de mise en commun des adresses MAC pour les cartes d'interface réseau (NIC) des machines virtuelles (VM).
daemonset/bridge-marker	Marque les ponts de réseau disponibles sur les nœuds en tant que ressources de nœud.
daemonset/kube-cni-linux-bridge-plugin	Installe des plugins CNI sur les nœuds du cluster, permettant d'attacher des VM à des ponts Linux par le biais de définitions d'attachement au réseau.

2.5. À PROPOS DE L'OPÉRATEUR HOSTPATH-PROVISIONER-OPERATOR

Le site **hostpath-provisioner-operator** déploie et gère le hostpath provisioner (HPP) multi-nœuds et les ressources associées.

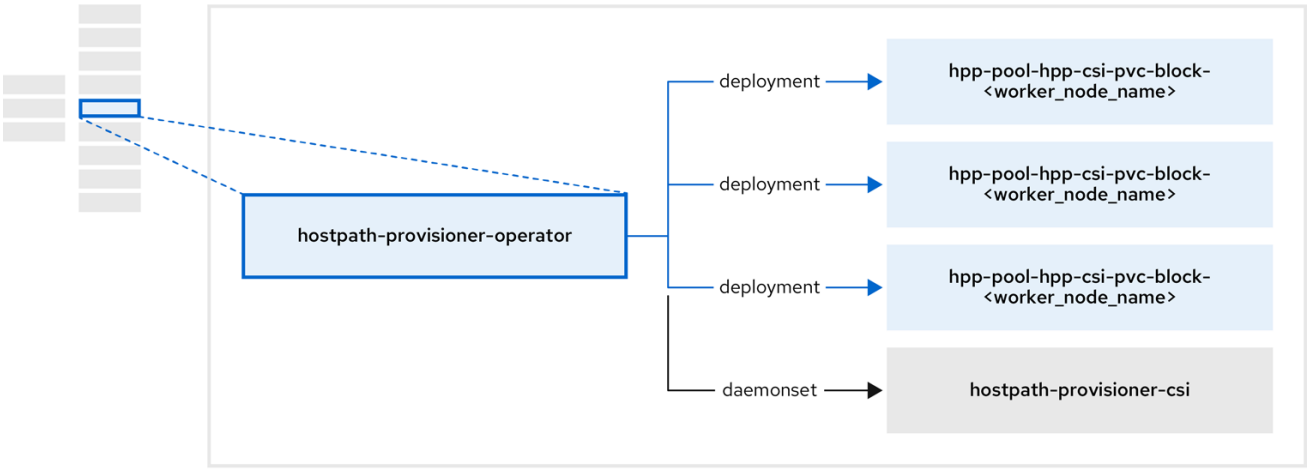


Tableau 2.4. composants hostpath-provisioner-operator

Component	Description
deployment/hpp-pool-hpp-csi-pvc-block-<worker_node_name>	Fournit un worker pour chaque nœud où le hostpath provisioner (HPP) est désigné pour s'exécuter. Les pods montent le stockage de secours spécifié sur le nœud.

Component	Description
daemonset/hostpath-provisioner-csi	Implémente l'interface du pilote de l'interface de stockage de conteneurs (CSI) du HPP.
daemonset/hostpath-provisioner	Implémente l'interface de pilotage héritée du HPP.

2.6. À PROPOS DE L'OPÉRATEUR SSP

Le site **ssp-operator** déploie les modèles communs, les sources de démarrage par défaut correspondantes et le validateur de modèles.

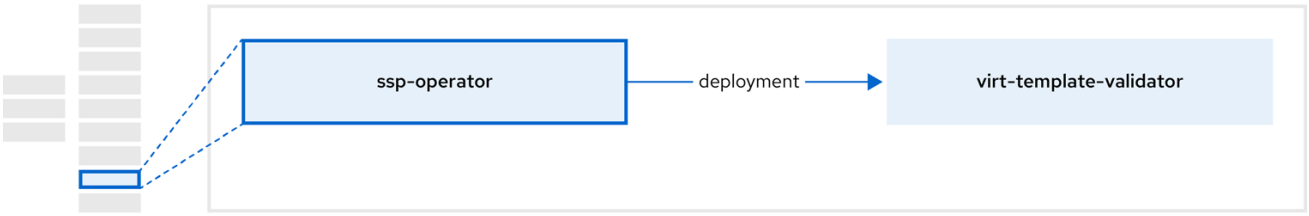


Tableau 2.5. composants de l'opérateur ssp

Component	Description
deployment/virt-template-validator	Vérifie les annotations **vm.kubevirt.io/validations** sur les machines virtuelles créées à partir de modèles et les rejette si elles sont invalides.

2.7. À PROPOS DE L'OPÉRATEUR TEKTON-TASKS

Le site **tekton-tasks-operator** déploie des exemples de pipelines montrant l'utilisation d'OpenShift Pipelines pour les VM. Il déploie également des tâches OpenShift Pipeline supplémentaires qui permettent aux utilisateurs de créer des VM à partir de modèles, de copier et de modifier des modèles, et de créer des volumes de données.

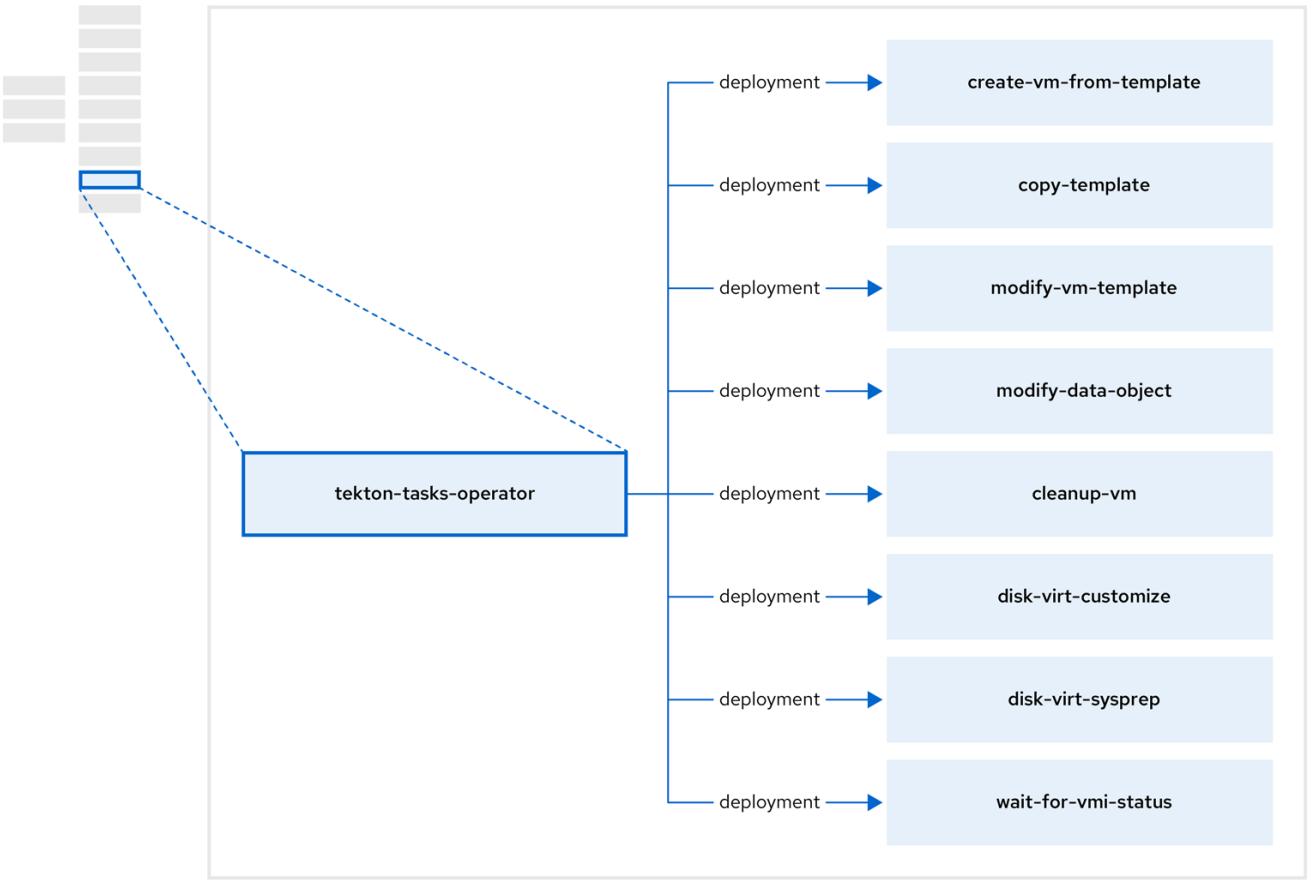


Tableau 2.6. composants tekton-tasks-operator

Component	Description
deployment/create-vm-from-template	Crée une VM à partir d'un modèle.
deployment/copy-template	Copie un modèle de VM.
deployment/modify-vm-template	Crée ou supprime un modèle de VM.
deployment/modify-data-object	Crée ou supprime des volumes de données ou des sources de données.
deployment/cleanup-vm	Exécute un script ou une commande sur une VM, puis arrête ou supprime la VM.
deployment/disk-virt-customize	Exécute un script **customize** sur un PVC cible en utilisant **virt-customize**.
deployment/disk-virt-sysprep	Exécute un script **sysprep** sur un PVC cible en utilisant **virt-sysprep**.
deployment/wait-for-vmi-status	Attend un statut VMI spécifique, puis échoue ou réussit en fonction de ce statut.

2.8. À PROPOS DE L'OPÉRATEUR VIRTUEL

Le site **virt-operator** déploie, met à niveau et gère OpenShift Virtualization sans perturber les charges de travail actuelles des machines virtuelles (VM).

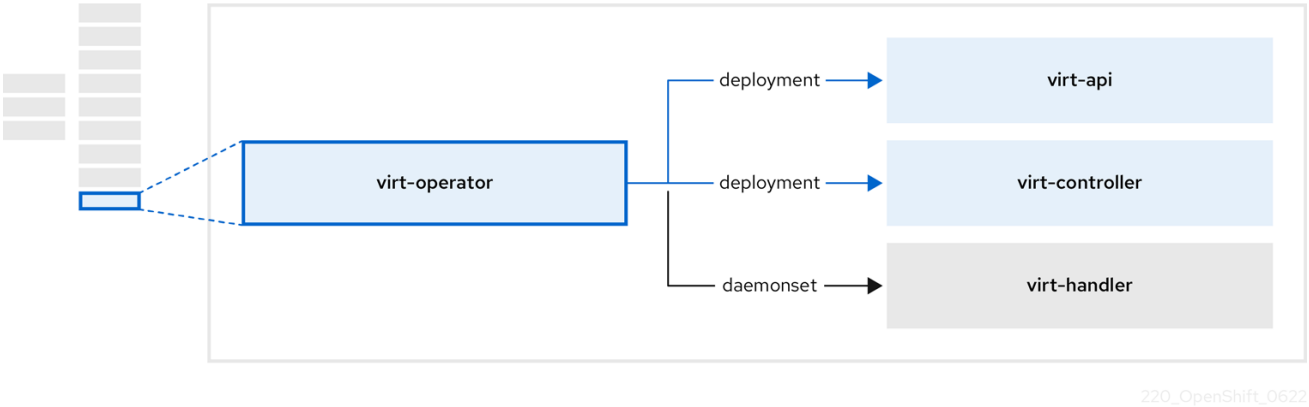


Tableau 2.7. composants de l'opérateur virtuel

Component	Description
deployment/virt-api	Serveur API HTTP qui sert de point d'entrée pour tous les flux liés à la virtualisation.
deployment/virt-controller	Observe la création d'un nouvel objet instance VM et crée un pod correspondant. Lorsque le pod est planifié sur un nœud, **virt-controller** met à jour la VM avec le nom du nœud.
daemonset/virt-handler	Surveille toute modification apportée à une VM et demande à **virt-launcher** d'effectuer les opérations requises. Ce composant est spécifique à chaque nœud.
pod/virt-launcher	Contient la VM créée par l'utilisateur, telle qu'elle est mise en œuvre par **libvirt** et **qemu**.

CHAPITRE 3. DÉMARRER AVEC OPENSHIFT VIRTUALIZATION

Vous pouvez explorer les caractéristiques et les fonctionnalités d'OpenShift Virtualization en installant et en configurant un environnement de base.

NOTE

Les procédures de configuration du cluster requièrent les privilèges de **cluster-admin**.

3.1. PLANIFIER ET INSTALLER OPENSHIFT VIRTUALIZATION

Planifier et installer OpenShift Virtualization sur un cluster OpenShift Container Platform :

- Planifiez votre cluster bare metal pour la virtualisation OpenShift .
- Préparez votre cluster pour la virtualisation OpenShift.
- Installer l'opérateur de virtualisation OpenShift .
- Installez l'outil d'interface de ligne de commande (CLI) **virtctl** .

Ressources pour la planification et l'installation

- Utilisation d'un fournisseur de stockage compatible CSI .
- Configuration du stockage local pour les machines virtuelles .
- Installation de l'opérateur NMState de Kubernetes .
- Spécification des nœuds pour les machines virtuelles .
- **Virtctl** commandes.

3.2. CRÉER ET GÉRER DES MACHINES VIRTUELLES

Créer des machines virtuelles (VM) en utilisant la console web :

- Créer rapidement une VM .
- Personnaliser un modèle pour créer une VM .

Se connecter aux machines virtuelles :

- Connectez-vous à la console série ou à la console VNC d'une VM à l'aide de la console web.
- Se connecter à une VM en utilisant SSH .
- Connectez-vous à une VM Windows en utilisant RDP .

Gérer les machines virtuelles :

- Arrêtez, démarrez, mettez en pause et redémarrez une VM à l'aide de la console web .
- Gérez une VM, exposez un port ou connectez-vous à la console série à l'aide de l'outil CLI **virtctl** .

3.3. PROCHAINES ÉTAPES

- Connectez les machines virtuelles aux réseaux secondaires :
  - Connecter une VM à un réseau pont Linux .
  - Connecter une VM à un réseau SR-IOV .

NOTE

Les VM sont connectées par défaut au réseau pod. Vous devez configurer un réseau secondaire, tel qu'un pont Linux ou un SR-IOV, puis ajouter le réseau à la configuration de la VM.

- Surveillez les ressources, les détails, l'état et les principaux consommateurs à l'aide de la console web.
- Affichez des informations de haut niveau sur les charges de travail des machines virtuelles en utilisant la console web.
- Visualiser les logs d'OpenShift Virtualization en utilisant le CLI .
- Automatiser les déploiements de VM Windows avec **sysprep**.
- Migrer en direct des machines virtuelles.
- Sauvegarde et restauration des machines virtuelles.

CHAPITRE 4. VUE D'ENSEMBLE DE LA CONSOLE WEB

La section **Virtualization** de la console web de OpenShift Container Platform contient les pages suivantes pour gérer et surveiller votre environnement de virtualisation OpenShift.

Tableau 4.1. Pages sur la virtualisation

Page	Description
Page de présentation	Gérer et surveiller l'environnement de virtualisation OpenShift.
Page de catalogue	Créer des machines virtuelles à partir d'un catalogue de modèles.
Page Machines virtuelles	Configurer et surveiller les machines virtuelles.
Page des modèles	Créer et gérer des modèles.
Page DataSources	Créer et gérer des DataSources pour les sources de démarrage de VirtualMachine.
Page MigrationPolicies	Créer et gérer des politiques de migration pour les charges de travail.

Tableau 4.2. Clé

Icône	Description
	Icône d'édition
	Icône de lien

4.1. PAGE DE PRÉSENTATION

La page Vue d'ensemble affiche les ressources, les mesures, la progression de la migration et les paramètres au niveau du cluster.

Exemple 4.1. Page de présentation

Élément	Description
Download virtctl 	Téléchargez l'outil de ligne de commande **virtctl** pour gérer les ressources.
Overview onglet	Ressources, utilisation, alertes et état.
Top consumers onglet	Les plus gros consommateurs de ressources de CPU, de mémoire et de stockage.

Élément	Description
Migrations onglet	Statut des migrations vivantes.
Settings onglet	Paramètres à l'échelle du cluster, y compris les limites de migration en direct et les autorisations des utilisateurs.

4.1.1. Onglet Vue d'ensemble

L'onglet **Overview** affiche les ressources, l'utilisation, les alertes et l'état.

Exemple 4.2. Overview onglet

Élément	Description
\Carte "Ressources de démarrage	• \Tuile "Démarrage rapide" : Apprenez à créer, importer et exécuter des machines virtuelles à l'aide d'instructions et de tâches étape par étape. • \Tuile "Feature highlights" : Lire les dernières informations sur les principales fonctionnalités de virtualisation. • \Tuile "Opérateurs connexes" : Installer des opérateurs tels que l'opérateur Kubernetes NMState ou l'opérateur OpenShift Data Foundation.
\Carreau "VirtualMachines	Nombre de machines virtuelles, avec un graphique montrant la tendance des 7 derniers jours.
\Utilisation de la vCPU" carreaux	l'utilisation des vCPU, avec un graphique montrant la tendance des 7 derniers jours.
\Carreau "Mémoire	Utilisation de la mémoire, avec un graphique montrant la tendance des 7 derniers jours.
\Dalle "Storage" (stockage)	Utilisation du stockage, avec un graphique montrant la tendance des 7 derniers jours.
\Carreau "Alerts	Alertes d'OpenShift Virtualization, regroupées par gravité.
\Carreau "VirtualMachine statuses	Nombre de machines virtuelles, regroupées par état.
\Graphique "Machines virtuelles par modèle	Nombre de machines virtuelles créées à partir de modèles, regroupées par nom de modèle.

4.1.2. Onglet Consommateurs supérieurs

L'onglet **Top consumers** affiche les principaux consommateurs de CPU, de mémoire et de stockage.

Exemple 4.3. Top consumers onglet

Élément	Description
View virtualization dashboard	Lien vers **Observe → Dashboards**, qui affiche les principaux consommateurs d'OpenShift Virtualization.
Time period liste	Sélectionnez une période pour filtrer les résultats.
Top consumers liste	Sélectionnez le nombre de premiers consommateurs pour filtrer les résultats.
\Graphique "CPU	Machines virtuelles dont l'utilisation du processeur est la plus élevée.
\Carte "Mémoire	Machines virtuelles utilisant le plus de mémoire.
\Tableau des échanges de mémoire	Machines virtuelles dont le trafic d'échange de mémoire est le plus élevé.
\Graphique "vCPU wait" (attente de vCPU)	Machines virtuelles ayant les périodes d'attente vCPU les plus élevées.
\Tableau "Stockage à travers le débit	Machines virtuelles dont le débit de stockage est le plus élevé.
\Graphique "Stockage IOPS	Machines virtuelles ayant le plus grand nombre d'opérations d'entrée/sortie de stockage par seconde.

4.1.3. Onglet Migrations

L'onglet **Migrations** affiche l'état des migrations de VirtualMachineInstance.

Exemple 4.4. Migrations onglet

Élément	Description
Time period liste	Sélectionnez une période pour filtrer les VirtualMachineInstanceMigrations.
VirtualMachineInstanceMigrations table	Liste des migrations de VirtualMachineInstance.

4.1.4. Onglet Paramètres

L'onglet **Settings** affiche les paramètres de l'ensemble du cluster dans les onglets suivants :

Tableau 4.3. Onglets sur Settings onglet

Onglet	Description
General onglet	Version d'OpenShift Virtualization et état des mises à jour.
Live migration onglet	Limites de migration et paramètres réseau en direct.
Templates project onglet	Projet pour les modèles Red Hat.
User permissions onglet	Autorisations pour les utilisateurs à l'échelle du cluster.

4.1.4.1. Onglet Général

L'onglet **General** affiche la version d'OpenShift Virtualization et l'état de la mise à jour.

Exemple 4.5. General onglet

Étiquette	Description
Nom du service	Virtualisation OpenShift
Fournisseur	Red Hat
Version installée	4.12.2
Mise à jour du statut	Exemple : **Up to date**
Chaîne	Canal sélectionné pour les mises à jour.

4.1.4.2. Onglet de migration en direct

Vous pouvez configurer la migration en direct dans l'onglet **Live migration**.

Exemple 4.6. Live migration onglet

Élément	Description
Max. migrations per cluster champ	Sélectionnez le nombre maximum de migrations en direct par cluster.

Élément	Description
Max. migrations per node champ	Sélectionnez le nombre maximal de migrations en direct par nœud.
Live migration network liste	Sélectionnez un réseau secondaire dédié pour la migration en direct.

4.1.4.3. Onglet Projet de modèles

Vous pouvez sélectionner un projet pour les modèles dans l'onglet **Templates project**.

Exemple 4.7. Templates project onglet

Élément	Description
Project liste	Sélectionnez un projet dans lequel stocker les modèles Red Hat. Le projet de modèles par défaut est **openshift**. Si vous souhaitez définir plusieurs projets de modèles, vous devez cloner les modèles de la page Modèles pour chaque projet.

4.1.4.4. Onglet Autorisations de l'utilisateur

L'onglet **User permissions** affiche les autorisations des utilisateurs à l'échelle du cluster pour les tâches.

Exemple 4.8. User permissions onglet

Élément	Description
User Permissions table	Liste des tâches, telles que **Share templates**, et des autorisations.

4.2. PAGE DE CATALOGUE

Vous pouvez créer une machine virtuelle en sélectionnant un modèle sur la page Catalogue.

Exemple 4.9. Page de catalogue

Élément	Description

Élément	Description
Templates project liste	Sélectionnez le projet dans lequel se trouvent vos modèles. Par défaut, les modèles Red Hat sont stockés dans le projet **openshift**. Vous pouvez modifier le projet de modèle dans l'ongletOverview → Settings → Template project .
All items|Default templates	Cliquez sur **Default templates** pour n'afficher que les modèles par défaut.
Boot source available case à cocher	Cochez la case pour afficher les modèles dont la source de démarrage est disponible.
Operating system cases à cocher	Cochez les cases pour afficher les modèles avec les systèmes d'exploitation sélectionnés.
Workload cases à cocher	Cochez les cases pour afficher les modèles avec les charges de travail sélectionnées.
Champ de recherche	Recherche de modèles par mot-clé.
Tuiles modèles	Cliquez sur une tuile de modèle pour afficher les détails du modèle et pour créer une machine virtuelle.

4.3. PAGE MACHINES VIRTUELLES

Vous pouvez créer et gérer des machines virtuelles sur la page Machines virtuelles.

Exemple 4.10. Page Machines virtuelles

Élément	Description
Create → From catalog	Créez une machine virtuelle sur la page Catalogue.
Create → With YAML	Créer une VirtualMachine en éditant un fichier de configuration YAML.
Filter champ	Filtrer les machines virtuelles par état, modèle, système d'exploitation ou nœud.
Champ de recherche	Recherchez les machines virtuelles par nom ou par étiquette.

Élément	Description
VirtualMachines table	Liste des machines virtuelles. Cliquez sur le menu Options  à côté d'une machine virtuelle pour sélectionner **Stop**, **Restart**, **Pause**, **Clone**, **Migrate**, **Copy SSH command**, **Edit labels**, **Edit annotations**, ou **Delete**. Cliquez sur une machine virtuelle pour accéder à la page de détails de la machine virtuelle.

4.3.1. Page de détail de VirtualMachine

Vous pouvez configurer une machine virtuelle sur la page de détails de la machine virtuelle.

Exemple 4.11. Page de détail de VirtualMachine

Élément	Description
Actions menu	Cliquez sur le menu **Actions** pour sélectionner **Stop**, **Restart**, **Pause**, **Clone**, **Migrate**, **Copy SSH command**, **Edit labels**, **Edit annotations**, ou **Delete**.
Overview onglet	Utilisation des ressources, alertes, disques et périphériques.
Details onglet	Configurations de machines virtuelles.
Metrics onglet	Mémoire, CPU, stockage, réseau et mesures de migration.
YAML onglet	Fichier de configuration YAML de VirtualMachine.
Scheduling onglet	Configurations d'ordonnancement.
Environment onglet	Gestion des cartes de configuration, des secrets et des comptes de service.
Events onglet	Flux d'événements de la machine virtuelle.
Console onglet	Gestion de la session de la console.
Network interfaces onglet	Gestion de l'interface réseau.
Disks onglet	Gestion des disques.

Élément	Description
Scripts onglet	Cloud-init et gestion des clés SSH.
Snapshots onglet	Gestion des instantanés.

4.3.1.1. Onglet Vue d'ensemble

L'onglet **Overview** affiche l'utilisation des ressources, les alertes et les informations de configuration.

Exemple 4.12. Onglet Vue d'ensemble

Élément	Description
\Carreau "Details	Informations générales sur la machine virtuelle.
\Carreau "Utilization	**CPU**, **Memory**, **Storage**, et **Network transfer**.
\Carreau "Dispositifs matériels	GPU et dispositifs hôtes.
\Carreau "Alerts	Alertes d'OpenShift Virtualization, regroupées par gravité.
\Carreau "Snapshots	**Take snapshot** et le tableau **Snapshots**.
\Carreau "Interfaces de réseau	**Network interfaces** table.
\Carreau "Disks	**Disks** table.

4.3.1.2. Onglet Détails

Vous pouvez configurer la machine virtuelle dans l'onglet **Details**.

Exemple 4.13. Onglet Détails

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Nom	Nom de la machine virtuelle.

Élément	Description
Espace de noms	L'espace de noms VirtualMachine.
Étiquettes	Cliquez sur l'icône d'édition pour modifier les étiquettes.
Annotations	Cliquez sur l'icône d'édition pour modifier les annotations.
Description	Cliquez sur l'icône de modification pour saisir une description.
Système d'exploitation	Nom du système d'exploitation.
CPU|Mémoire	Cliquez sur l'icône d'édition pour modifier la requête CPU|Mémoire. The number of CPUs is calculated by using the following formula: **sockets * threads * cores**.
Type de machine	Type de machine VirtualMachine.
Mode d'amorçage	Cliquez sur l'icône d'édition pour modifier le mode de démarrage.
Démarrage en mode pause	Cliquez sur l'icône de modification pour activer ce paramètre.
Modèle	Nom du modèle utilisé pour créer la machine virtuelle.
Créé à	Date de création de la machine virtuelle.
Propriétaire	Propriétaire de la machine virtuelle.
Statut	Statut de la machine virtuelle.
Cosse	**virt-launcher** nom du pod.
VirtualMachineInstance	Nom de l'instance de la machine virtuelle.
Boot order	Cliquez sur l'icône d'édition pour sélectionner une source de démarrage.
IP address	Adresse IP de la machine virtuelle.
Nom d'hôte	Nom d'hôte de la machine virtuelle.
Fuseau horaire	Fuseau horaire de la machine virtuelle.
Nœud	Nœud sur lequel s'exécute la machine virtuelle.
Profil de la charge de travail	Cliquez sur l'icône d'édition pour modifier le profil de charge de travail.

Élément	Description
SSH à l'aide de virtctl	Cliquez sur l'icône de copie pour copier la commande **virtctl ssh** dans le presse-papiers.
SSH sur NodePort	La sélection de **Create a Service to expose your VirtualMachine for SSH access** génère une commande **ssh -p <port>**. Cliquez sur l'icône de copie pour copier la commande dans le presse-papiers.
Dispositifs GPU	Cliquez sur l'icône de modification pour ajouter un appareil GPU.
Dispositifs hôtes	Cliquez sur l'icône de modification pour ajouter un périphérique hôte.
Section des services	Voir les services.
Section des utilisateurs actifs	Voir les utilisateurs actifs.

4.3.1.3. Onglet Métriques

L'onglet **Metrics** affiche les graphiques d'utilisation de la mémoire, du processeur, du stockage, du réseau et de la migration.

Exemple 4.14. Onglet Métriques

Élément	Description
Time range liste	Sélectionnez une période pour filtrer les résultats.
Virtualization dashboard	Lien vers l'onglet **Workloads** du projet en cours.
Section utilisation	**Memory**, **CPU** et **Network interface**.
Espace de stockage	**Storage total read/write** et **Storage iops total read/write**.
Section réseau	**Network in**, **Network out** et **Network bandwidth**.
Section migration	**Migration** et **KV data transfer rate**.

4.3.1.4. Onglet YAML

Vous pouvez configurer la machine virtuelle en modifiant le fichier YAML dans l'onglet **YAML**.

Exemple 4.15. Onglet YAML

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Save bouton	Enregistrer les modifications dans le fichier YAML.
Reload bouton	Annulez vos modifications et rechargez le fichier YAML.
Cancel bouton	Quittez l'onglet **YAML**.
Download bouton	Téléchargez le fichier YAML sur votre machine locale.

4.3.1.5. Onglet Programmation

Vous pouvez configurer la planification dans l'onglet **Scheduling**.

Exemple 4.16. Onglet Programmation

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Sélecteur de nœuds	Cliquez sur l'icône d'édition pour ajouter une étiquette afin de spécifier les nœuds de qualification.
Tolérances	Cliquez sur l'icône d'édition pour ajouter une tolérance afin de spécifier les nœuds de qualification.
Règles d'affinité	Cliquez sur l'icône de modification pour ajouter une règle d'affinité.
Descheduler interrupteur	Activer ou désactiver l'ordonnanceur. L'ordonnanceur évince un pod en cours d'exécution afin qu'il puisse être reprogrammé sur un nœud plus approprié.
Ressources dédiées	Cliquez sur l'icône d'édition pour sélectionner **Schedule this workload with dedicated resources (guaranteed policy)**.
Stratégie d'expulsion	Cliquez sur l'icône d'édition pour sélectionner **LiveMigrate** comme stratégie d'éviction de VirtualMachineInstance.

4.3.1.6. Onglet Environnement

Vous pouvez gérer les cartes de configuration, les secrets et les comptes de service dans l'onglet **Environment**.

Exemple 4.17. Onglet Environnement

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Add Config Map, Secret or Service Account	Cliquez sur le lien et sélectionnez une carte de configuration, un secret ou un compte de service dans la liste des ressources.

4.3.1.7. Onglet Événements

L'onglet **Events** affiche une liste des événements de VirtualMachine.

4.3.1.8. Onglet Console

Vous pouvez ouvrir une session de console sur la machine virtuelle dans l'onglet **Console**.

Exemple 4.18. Onglet Console

Élément	Description
Section des informations d'identification de l'invité	Développez **Guest login credentials** pour afficher les informations d'identification créées avec **cloud-init**. Cliquez sur l'icône de copie pour copier les informations d'identification dans le presse-papiers.
Console liste	Sélectionnez **VNC console** ou **Serial console**. Vous pouvez sélectionner **Desktop viewer** pour vous connecter aux machines virtuelles Windows en utilisant le protocole Remote Desktop Protocol (RDP). Vous devez installer un client RDP sur une machine du même réseau.
Send key liste	Sélectionnez une combinaison de touches à envoyer à la console.
Disconnect bouton	Déconnecter la connexion de la console. Vous devez déconnecter manuellement la connexion à la console si vous ouvrez une nouvelle session de console. Dans le cas contraire, la première session de console continue de s'exécuter en arrière-plan.

4.3.1.9. Onglet Interfaces réseau

Vous pouvez gérer les interfaces réseau dans l'onglet **Network interfaces**.

Exemple 4.19. Onglet Interfaces réseau

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Add network interface bouton	Ajouter une interface réseau à la machine virtuelle.
Filter champ	Filtrer par type d'interface.
Champ de recherche	Recherche d'une interface réseau par nom ou par étiquette.
Network interface table	Liste des interfaces réseau. Cliquez sur le menu Options  à côté d'une interface réseau pour sélectionner **Edit** ou **Delete**.

4.3.1.10. Onglet Disques

Vous pouvez gérer les disques dans l'onglet **Disks**.

Exemple 4.20. Onglet Disques

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Add disk bouton	Ajouter un disque à la machine virtuelle.
Filter champ	Filtrer par type de disque.
Champ de recherche	Recherche d'un disque par son nom.
Disks table	Liste des disques de la machine virtuelle. Cliquez sur le menu Options  à côté d'un disque pour sélectionner **Edit** ou **Detach**.

Paramètres	Description
File systems table	Liste des systèmes de fichiers de VirtualMachine.

4.3.1.11. Onglet Scripts

Vous pouvez gérer les clés cloud-init et SSH de la machine virtuelle dans l'onglet **Scripts**.

Exemple 4.21. Onglet Scripts

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Cloud-init	Cliquez sur l'icône d'édition pour modifier les paramètres de démarrage du nuage.
Clé SSH autorisée	Cliquez sur l'icône de modification pour créer un nouveau secret ou pour joindre un secret existant.

4.3.1.12. Onglet Instantanés

Vous pouvez créer des instantanés et restaurer des machines virtuelles à partir d'instantanés dans l'onglet **Snapshots**.

Exemple 4.22. Onglet Instantanés

Élément	Description
Take snapshot bouton	Créer un instantané.
Filter champ	Filtrer les instantanés par état.
Champ de recherche	Recherche d'instantanés par nom ou par étiquette.
Snapshot table	Liste des instantanés. Cliquez sur le menu Options  à côté d'un instantané pour sélectionner **Edit labels**, **Edit annotations**, **Edit VirtualMachineSnapshot**, **Delete VirtualMachineSnapshot**.

4.4. PAGE DES MODÈLES

Vous pouvez créer, modifier et cloner des modèles de machine virtuelle sur la page Modèles.

NOTE

Vous ne pouvez pas modifier un modèle Red Hat. Vous pouvez cloner un modèle Red Hat et le modifier pour créer un modèle personnalisé.

Exemple 4.23. Page des modèles

Élément	Description
Create Template bouton	Créer un modèle en éditant un fichier de configuration YAML.
Filter champ	Filtrer les modèles par type, source de démarrage, fournisseur de modèles ou système d'exploitation.
Champ de recherche	Recherchez des modèles par nom ou par étiquette.
Templates table	Liste des modèles. Cliquez sur le menu Options  à côté d'un modèle pour sélectionner **Edit**, **Clone**, **Edit boot source**, **Edit boot source reference**, **Edit labels**, **Edit annotations**, ou **Delete**.

4.4.1. Page de détails du modèle

Vous pouvez visualiser les paramètres du modèle et modifier les modèles personnalisés sur la page Détails du modèle.

Exemple 4.24. Page de détails du modèle

Élément	Description
Actions menu	Cliquez sur le menu **Actions** pour sélectionner **Edit**, **Clone**, **Edit boot source**, **Edit boot source reference**, **Edit labels**, **Edit annotations**, ou **Delete**.
Details onglet	Paramètres et configurations des modèles.
YAML onglet	Fichier de configuration YAML.
Scheduling onglet	Configurations d'ordonnancement.

Élément	Description
Network interfaces onglet	Gestion de l'interface réseau.
Disks onglet	Gestion des disques.
Scripts onglet	Gestion du Cloud-init, des clés SSH et de Sysprep.
Parameters onglet	Paramètres.

4.4.1.1. Onglet Détails

Vous pouvez configurer un modèle personnalisé dans l'onglet **Details**.

Exemple 4.25. Onglet Détails

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Nom	Nom du modèle.
Espace de noms	Espace de noms du modèle.
Étiquettes	Cliquez sur l'icône d'édition pour modifier les étiquettes.
Annotations	Cliquez sur l'icône d'édition pour modifier les annotations.
Nom d'affichage	Cliquez sur l'icône d'édition pour modifier le nom d'affichage.
Description	Cliquez sur l'icône de modification pour saisir une description.
Système d'exploitation	Nom du système d'exploitation.
CPU|Mémoire	Cliquez sur l'icône d'édition pour modifier la requête CPU|Mémoire. The number of CPUs is calculated by using the following formula: **sockets * threads * cores**.
Type de machine	Type de machine à gabarit.
Mode d'amorçage	Cliquez sur l'icône d'édition pour modifier le mode de démarrage.
Modèle de base	Nom du modèle de base utilisé pour créer ce modèle.

Élément	Description
Créé à	Date de création du modèle.
Propriétaire	Propriétaire du modèle.
Boot order	Modèle de commande de démarrage.
Source de démarrage	Disponibilité de la source de démarrage.
Fournisseur	Fournisseur de modèles.
Soutien	Niveau de soutien du modèle.
Dispositifs GPU	Cliquez sur l'icône de modification pour ajouter un appareil GPU.
Dispositifs hôtes	Cliquez sur l'icône de modification pour ajouter un périphérique hôte.

4.4.1.2. Onglet YAML

Vous pouvez configurer un modèle personnalisé en modifiant le fichier YAML dans l'onglet **YAML**.

Exemple 4.26. Onglet YAML

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Save bouton	Enregistrer les modifications dans le fichier YAML.
Reload bouton	Annulez vos modifications et rechargez le fichier YAML.
Cancel bouton	Quittez l'onglet **YAML**.
Download bouton	Téléchargez le fichier YAML sur votre machine locale.

4.4.1.3. Onglet Programmation

Vous pouvez configurer la planification dans l'onglet **Scheduling**.

Exemple 4.27. Onglet Programmation

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Sélecteur de nœuds	Cliquez sur l'icône d'édition pour ajouter une étiquette afin de spécifier les nœuds de qualification.
Tolérances	Cliquez sur l'icône d'édition pour ajouter une tolérance afin de spécifier les nœuds de qualification.
Règles d'affinité	Cliquez sur l'icône de modification pour ajouter une règle d'affinité.
Descheduler interrupteur	Activer ou désactiver l'ordonnanceur. L'ordonnanceur évince un pod en cours d'exécution afin qu'il puisse être reprogrammé sur un nœud plus approprié.
Ressources dédiées	Cliquez sur l'icône d'édition pour sélectionner **Schedule this workload with dedicated resources (guaranteed policy)**.
Stratégie d'expulsion	Cliquez sur l'icône d'édition pour sélectionner **LiveMigrate** comme stratégie d'éviction de VirtualMachineInstance.

4.4.1.4. Onglet Interfaces réseau

Vous pouvez gérer les interfaces réseau dans l'onglet **Network interfaces**.

Exemple 4.28. Onglet Interfaces réseau

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Add network interface bouton	Ajouter une interface réseau au modèle.
Filter champ	Filtrer par type d'interface.
Champ de recherche	Recherche d'une interface réseau par nom ou par étiquette.
Network interface table	Liste des interfaces réseau. Cliquez sur le menu Options  à côté d'une interface réseau pour sélectionner **Edit** ou **Delete**.

4.4.1.5. Onglet Disques

Vous pouvez gérer les disques dans l'onglet **Disks**.

Exemple 4.29. Onglet Disques

Paramètres	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Add disk bouton	Ajouter un disque au modèle.
Filter champ	Filtrer par type de disque.
Champ de recherche	Recherche d'un disque par son nom.
Disks table	Liste des disques modèles. Cliquez sur le menu Options  à côté d'un disque pour sélectionner **Edit** ou **Detach**.

4.4.1.6. Onglet Scripts

Vous pouvez gérer les paramètres de démarrage du nuage, les clés SSH et les fichiers de réponse Sysprep dans l'onglet **Scripts**.

Exemple 4.30. Onglet Scripts

Élément	Description
YAML interrupteur	Réglez sur **ON** pour voir vos modifications en direct dans le fichier de configuration YAML.
Cloud-init	Cliquez sur l'icône d'édition pour modifier les paramètres de démarrage du nuage.
Clé SSH autorisée	Cliquez sur l'icône de modification pour créer un nouveau secret ou pour joindre un secret existant.
Sysprep	Cliquez sur l'icône d'édition pour télécharger un fichier de réponse **Autounattend.xml** ou **Unattend.xml** afin d'automatiser l'installation de Windows VirtualMachine.

4.4.1.7. Onglet Paramètres

Vous pouvez modifier les paramètres des modèles sélectionnés dans l'onglet **Parameters**.

Exemple 4.31. Onglet Paramètres

Élément	Description
Nom de la VM	Sélectionnez **Generated (expression)** pour une valeur générée, **Value** pour définir une valeur par défaut ou **None** dans la liste **Default value type**.
Espace de noms de la source de données	Sélectionnez **Generated (expression)** pour une valeur générée, **Value** pour définir une valeur par défaut ou **None** dans la liste **Default value type**.
Mot de passe de l'utilisateur du cloud	Sélectionnez **Generated (expression)** pour une valeur générée, **Value** pour définir une valeur par défaut ou **None** dans la liste **Default value type**.

4.5. PAGE DATASOURCES

Vous pouvez créer et configurer des sources de données pour les sources de démarrage de la machine virtuelle sur la page DataSources.

Lorsque vous créez une source de données, une ressource **DataImportCron** définit une tâche cron pour interroger et importer l'image disque, sauf si vous désactivez les mises à jour automatiques de la source de démarrage.

Exemple 4.32. Page DataSources

Élément	Description
Create DataSource → With form	Créez une source de données en saisissant l'URL du registre, la taille du disque, le nombre de révisions et l'expression cron dans un formulaire.
Create DataSources → With YAML	Créez une source de données en éditant un fichier de configuration YAML.
Filter champ	Filtrer les DataSources en fonction d'attributs tels que DataImportCron disponible.
Champ de recherche	Recherche d'une source de données par nom ou par étiquette.
DataSources table	Liste des sources de données. Cliquez sur le menu Options  à côté d'une source de données pour sélectionner **Edit labels**, **Edit annotations**, ou **Delete**.

Cliquez sur une source de données pour afficher la page de détails de la source de données.

4.5.1. Page de détails sur les sources de données

Vous pouvez configurer une source de données sur la page de détails de la source de données.

Exemple 4.33. Page de détails sur les sources de données

Élément	Description
Details onglet	Configurer une source de données en modifiant un formulaire.
YAML onglet	Configurer une source de données en éditant un fichier de configuration YAML.
Actions menu	Sélectionnez **Edit labels**, **Edit annotations**, ou **Delete**.
Nom	Nom de la source de données.
Espace de noms	Espace de noms DataSource.
Étiquettes	Cliquez sur l'icône d'édition pour modifier les étiquettes.
Annotations	Cliquez sur l'icône d'édition pour modifier les annotations.
Conditions	Affiche les conditions d'état de la source de données.

4.6. PAGE MIGRATIONPOLICIES

Vous pouvez gérer les politiques de migration pour vos charges de travail sur la page Politiques de migration.

Exemple 4.34. Page MigrationPolicies

Élément	Description
Create MigrationPolicy → With form	Créez une MigrationPolicy en saisissant des configurations et des étiquettes dans un formulaire.
Create MigrationPolicy → With YAML	Créer une MigrationPolicy en éditant un fichier de configuration YAML.
Name | Label champ de recherche	Recherche d'une MigrationPolicy par nom ou par étiquette.

Élément	Description
MigrationPolicies table	Liste des politiques de migration. Cliquez sur le menu Options  à côté d'une Politique de migration pour sélectionner **Edit** ou **Delete**.

Cliquez sur une Politique de migration pour afficher la page de détails de la Politique de migration.

4.6.1. Page de détails de la politique de migration

Vous pouvez configurer une MigrationPolicy sur la page de détails de la MigrationPolicy.

Exemple 4.35. Page de détails de la politique de migration

Élément	Description
Details onglet	Configurer une MigrationPolicy en éditant un formulaire.
YAML onglet	Configurer une MigrationPolicy en éditant un fichier de configuration YAML.
Actions menu	Sélectionnez **Edit** ou **Delete**.
Nom	Nom de la politique de migration.
Description	Description de la politique de migration.
Configurations	Cliquez sur l'icône d'édition pour mettre à jour les configurations de la Politique de migration.
Largeur de bande par migration	Demande de bande passante par migration. Pour une bande passante illimitée, réglez la valeur sur **0**.
Convergence automatique	Politique de convergence automatique.
Post-copie	Politique de post-copie.
Délai d'achèvement	Valeur du délai d'achèvement en secondes.
Étiquettes du projet	Cliquez sur **Edit** pour modifier les étiquettes du projet.

Élément	Description
Etiquettes VirtualMachine	Cliquez sur **Edit** pour modifier les étiquettes de la machine virtuelle.

CHAPITRE 5. NOTES DE VERSION D'OPENSHIFT VIRTUALIZATION

5.1. RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

5.2. À PROPOS DE LA VIRTUALISATION RED HAT OPENSHIFT

Red Hat OpenShift Virtualization vous permet d'amener des machines virtuelles (VM) traditionnelles dans OpenShift Container Platform où elles s'exécutent aux côtés des conteneurs et sont gérées comme des objets Kubernetes natifs.

OpenShift Virtualization est représenté par le module  .

Vous pouvez utiliser OpenShift Virtualization avec le fournisseur de réseau OVN-Kubernetes ou le fournisseur de réseau Container Network Interface (CNI) par défaut d'OpenShiftSDN.

En savoir plus sur ce que vous pouvez faire avec OpenShift Virtualization .

En savoir plus sur l'architecture et les déploiements d'OpenShift Virtualization .

Préparez votre cluster pour la virtualisation OpenShift.

5.2.1. Version de cluster prise en charge par OpenShift Virtualization

OpenShift Virtualization 4.12 est pris en charge pour une utilisation sur les clusters OpenShift Container Platform 4.12. Pour utiliser la dernière version z-stream d'OpenShift Virtualization, vous devez d'abord passer à la dernière version d'OpenShift Container Platform.

5.2.2. Systèmes d'exploitation invités pris en charge

Pour afficher les systèmes d'exploitation invités pris en charge pour OpenShift Virtualization, reportez-vous à Systèmes d'exploitation invités certifiés dans Red Hat OpenStack Platform, Red Hat Virtualization et OpenShift Virtualization.

5.3. CARACTÉRISTIQUES NOUVELLES ET MODIFIÉES

- OpenShift Virtualization est certifié dans le cadre du programme de validation de la virtualisation de Windows Server (SVVP) de Microsoft pour exécuter des charges de travail Windows Server.
  La certification SVVP s'applique à :
  - Les travailleurs de Red Hat Enterprise Linux CoreOS. Dans le catalogue SVVP de Microsoft, ils sont nommés *Red Hat OpenShift Container Platform 4 on RHEL CoreOS 8* .
  - CPU Intel et AMD.

- OpenShift Virtualization n'utilise plus l'option  n'est plus utilisé. OpenShift Virtualization est désormais représenté par le logo  pour les versions 4.9 et suivantes.

- Vous pouvez créer un vidage de la mémoire de la VM à des fins d'analyse judiciaire à l'aide de la commande**virtctl memory-dump** .

- Vous pouvez exporter et télécharger un volume à partir d'une machine virtuelle (VM), d'un instantané de VM ou d'une revendication de volume persistant (PVC) pour le recréer sur un cluster différent ou dans un espace de noms différent sur le même cluster à l'aide de la commande **virtctl vmexport** ou en créant une ressource personnalisée **VirtualMachineExport**. Vous pouvez également exporter le vidage de la mémoire à des fins d'analyse médico-légale.

- Vous pouvez découvrir les fonctions et l'organisation de la console web d'OpenShift Virtualization en vous référant à la documentation de présentation de la console web .

- Vous pouvez utiliser la commande **virtctl ssh** pour transférer le trafic SSH vers une machine virtuelle en utilisant votre client SSH local ou en copiant la commande SSH depuis la console web d'OpenShift Container Platform.

- Les volumes de données autonomes et les volumes de données créés lors de l'utilisation de **dataVolumeTemplate** pour préparer un disque pour une VM ne sont plus stockés dans le système. Les volumes de données sont désormais automatiquement ramassés et supprimés après la création du PVC.

- OpenShift Virtualization fournit désormais des mesures de migration en direct auxquelles vous pouvez accéder en utilisant le tableau de bord de surveillance d'OpenShift Container Platform.

- L'opérateur de virtualisation OpenShift lit maintenant le profil de sécurité TLS à l'échelle du cluster à partir de la ressource personnalisée **APIServer** et le propage aux composants de virtualisation OpenShift, y compris la virtualisation, le stockage, le réseau et l'infrastructure.

- OpenShift Virtualization dispose de runbooks pour vous aider à résoudre les problèmes qui déclenchent des alertes. Les alertes sont affichées sur la page **Virtualization → Overview** de la console web. Chaque runbook définit une alerte et fournit des étapes pour diagnostiquer et résoudre le problème. Cette fonctionnalité a été introduite précédemment en tant qu'aperçu technologique et est maintenant disponible de manière générale.

5.3.1. Démarrage rapide

- Des visites de démarrage rapide sont disponibles pour plusieurs fonctionnalités d'OpenShift Virtualization. Pour afficher les visites, cliquez sur l'icône **Help ?** dans la barre de menu de l'en-tête de la console OpenShift Virtualization, puis sélectionnez **Quick Starts**. Vous pouvez filtrer les visites disponibles en saisissant le mot-clé **virtualization** dans le champ **Filter**.

5.3.2. Mise en réseau

- Vous pouvez maintenant spécifier l'espace de noms dans lequel le contrôle du cluster OpenShift Container Platform doit être exécuté.

- Vous pouvez maintenant configurer un service de répartition de charge en utilisant l' opérateur MetalLB en mode couche 2.

5.3.3. Console web

- La page **Virtualization → Overview** présente les améliorations suivantes en termes de convivialité :
  - Un lien **Download virtctl** est disponible.
  - Les informations sur les ressources sont personnalisées pour les utilisateurs administratifs et non administratifs. Par exemple, les utilisateurs non administratifs ne voient que leurs machines virtuelles.
  - L'onglet **Overview** affiche le nombre de VM et l'utilisation des vCPU, de la mémoire et du stockage avec des graphiques montrant la tendance des 7 derniers jours.
  - La carte **Alerts** de l'onglet **Overview** affiche les alertes regroupées par gravité.
  - L'onglet **Top Consumers** affiche les principaux consommateurs de CPU, de mémoire et de stockage sur une période de temps configurable.
  - L'onglet **Migrations** affiche la progression des migrations de VM.
  - L'onglet **Settings** affiche les paramètres de l'ensemble du cluster, y compris les limites de migration en direct, le réseau de migration en direct et le projet de modèles.
- Vous pouvez créer et gérer des politiques de migration en direct en un seul endroit sur la page **Virtualization → MigrationPolicies**.
- L'onglet **Metrics** de la page **VirtualMachine details** affiche les mesures de mémoire, de CPU, de stockage, de réseau et de migration d'une VM, sur une période de temps configurable.
- Lorsque vous personnalisez un modèle pour créer une VM, vous pouvez définir le commutateur **YAML** sur **ON** dans chaque onglet de configuration de la VM pour afficher les modifications en direct dans le fichier de configuration YAML à côté du formulaire.
- L'onglet **Migrations** de la page **Virtualization → Overview** affiche la progression des migrations d'instances de machines virtuelles sur une période de temps configurable.
- Vous pouvez désormais définir un réseau dédié pour la migration en direct afin de minimiser les interruptions des charges de travail des locataires. Pour sélectionner un réseau, accédez à **Virtualization → Overview → Settings → Live migration**.

5.3.4. Fonctionnalités obsolètes

Les fonctionnalités obsolètes sont incluses dans la version actuelle et sont prises en charge. Cependant, elles seront supprimées dans une prochaine version et ne sont pas recommandées pour les nouveaux déploiements.

5.3.5. Fonctionnalités supprimées

Les fonctionnalités supprimées ne sont pas prises en charge dans la version actuelle.

- La prise en charge de l'ancienne ressource personnalisée HPP, et de la classe de stockage associée, a été supprimée pour tous les nouveaux déploiements. Dans OpenShift Virtualization 4.12, l'opérateur HPP utilise le pilote Kubernetes Container Storage Interface (CSI) pour configurer le stockage local. Une ressource personnalisée HPP héritée est prise en charge uniquement si elle a été installée sur une version précédente d'OpenShift Virtualization.

- OpenShift Virtualization 4.11 a supprimé la prise en charge de nmstate, y compris les objets suivants :
  - **NodeNetworkState**
  - **NodeNetworkConfigurationPolicy**
  - **NodeNetworkConfigurationEnactment**

  Pour préserver et prendre en charge votre configuration nmstate existante, installez l'opérateur Kubernetes NMState avant de mettre à jour OpenShift Virtualization 4.11. Pour les versions 4.12 pour Extended Update Support (EUS), installez le Kubernetes NMState Operator après la mise à jour vers la version 4.12. Vous pouvez installer l'opérateur à partir de **OperatorHub** dans la console web d'OpenShift Container Platform, ou en utilisant l'OpenShift CLI (**oc**).

- Le Node Maintenance Operator (NMO) n'est plus livré avec OpenShift Virtualization. Vous pouvez installer le NMO à partir de **OperatorHub** dans la console web d'OpenShift Container Platform, ou en utilisant la CLI d'OpenShift (**oc**).
  Vous devez effectuer l'une des tâches suivantes avant de mettre à jour OpenShift Virtualization 4.11 à partir de OpenShift Virtualization 4.10.2 et des versions 4.10 ultérieures. Pour les versions Extended Update Support (EUS), vous devez effectuer les tâches suivantes avant de mettre à jour OpenShift Virtualization 4.12 à partir de la version 4.10.2 et des versions 4.10 ultérieures :
  - Sortir tous les nœuds du mode maintenance.
  - Installez l'ONM autonome et remplacez la ressource personnalisée (CR) **nodemaintenances.nodemaintenance.kubevirt.io** par une CR **nodemaintenances.nodemaintenance.medik8s.io**.

5.4. CARACTÉRISTIQUES DE L'APERÇU TECHNOLOGIQUE

Certaines fonctionnalités de cette version sont actuellement en avant-première technologique. Ces fonctionnalités expérimentales ne sont pas destinées à une utilisation en production. Notez l'étendue de l'assistance suivante sur le portail client de Red Hat pour ces fonctionnalités :

Aperçu de la technologie Fonctionnalités Support Champ d'application

- Vous pouvez désormais exécuter des vérifications de cluster OpenShift Container Platform pour mesurer la latence du réseau entre les VM.
- L'opérateur de tâches Tekton (TTO) intègre désormais la virtualisation OpenShift avec Red Hat OpenShift Pipelines. TTO inclut des tâches de cluster et des exemples de pipelines qui vous permettent de :
  - Créer et gérer des machines virtuelles (VM), des réclamations de volumes persistants (PVC) et des volumes de données.
  - Exécuter des commandes dans des machines virtuelles.
  - Manipuler les images de disques avec les outils de **libguestfs**.
  - Installer Windows 10 dans un nouveau volume de données à partir d'une image d'installation de Windows (fichier ISO).
  - Personnalisez une installation de base de Windows 10, puis créez une nouvelle image et un nouveau modèle.

- Vous pouvez maintenant utiliser la sonde ping de l'agent invité pour déterminer si l'agent invité QEMU fonctionne sur une machine virtuelle.

- Vous pouvez désormais utiliser Microsoft Windows 11 comme système d'exploitation invité. Cependant, OpenShift Virtualization 4.12 ne prend pas en charge les disques USB, qui sont nécessaires pour une fonction critique de la récupération BitLocker. Pour protéger les clés de récupération, utilisez d'autres méthodes décrites dans le guide de récupération BitLocker.

- Vous pouvez créer des stratégies de migration en direct avec des paramètres spécifiques, tels que l'utilisation de la bande passante, le nombre maximum de migrations parallèles et le délai d'attente, et appliquer les stratégies à des groupes de machines virtuelles en utilisant des étiquettes de machines virtuelles et d'espaces de noms.

5.5. BUG FIXES

- Vous pouvez désormais configurer le CR **HyperConverged** pour activer les périphériques à médiation avant l'installation des pilotes sans perdre la nouvelle configuration des périphériques après l'installation des pilotes. (BZ#2046298)

- Le fournisseur de réseau du cluster OVN-Kubernetes ne se bloque plus en raison de l'utilisation maximale de la RAM et du CPU si vous créez un grand nombre de services **NodePort**. (OCPBUGS-1940)

- Le clonage de plus de 100 VM à la fois n'échoue plus de manière intermittente si vous utilisez Red Hat Ceph Storage ou Red Hat OpenShift Data Foundation Storage. (BZ#1989527)

5.6. PROBLÈMES CONNUS

- Dans un cluster hétérogène avec différents nœuds de calcul, les machines virtuelles pour lesquelles HyperV Reenlightenment est activé ne peuvent pas être planifiées sur des nœuds qui ne prennent pas en charge la mise à l'échelle du compteur d'horodatage (TSC) ou qui ont la fréquence TSC appropriée. (BZ#2151169)

- Lorsque vous utilisez deux pods avec des contextes SELinux différents, les VM avec la classe de stockage **ocs-storagecluster-cephfs** ne parviennent pas à migrer et l'état de la VM devient **Paused**. Cela est dû au fait que les deux pods tentent d'accéder au volume CephFS **ReadWriteMany** partagé en même temps. (BZ#2092271)

    - Comme solution de contournement, utilisez la classe de stockage **ocs-storagecluster-ceph-rbd** pour migrer en direct des VM sur un cluster qui utilise Red Hat Ceph Storage.

- La chaîne de nom du provisionneur **TopoLVM** a changé dans OpenShift Virtualization 4.12. Par conséquent, l'importation automatique d'images de systèmes d'exploitation peut échouer avec le message d'erreur suivant (BZ#2158521) :

    ```
    DataVolume.storage spec is missing accessMode and volumeMode, cannot get access mode from StorageProfile.
    ```

    - En guise de solution de rechange :

        1. Mettre à jour le tableau **claimPropertySets** du profil de stockage :

            ```
            $ oc patch storageprofile <storage_profile> --type=merge -p '{"spec": {"claimPropertySets": [{"accessModes": ["ReadWriteOnce"], "volumeMode": "Block"}, \
            ```

```
{"accessModes": ["ReadWriteOnce"], "volumeMode": "Filesystem"}]}}'
```

2. Supprimez les volumes de données concernés dans l'espace de noms **openshift-virtualization-os-images**. Ils sont recréés avec le mode d'accès et le mode de volume du profil de stockage mis à jour.

- Lors de la restauration d'un instantané de VM pour un stockage dont le mode de liaison est **WaitForFirstConsumer**, les PVC restaurés restent dans l'état **Pending** et l'opération de restauration ne progresse pas.
  - Pour contourner le problème, démarrez la VM restaurée, arrêtez-la, puis redémarrez-la. La VM sera planifiée, les PVC seront dans l'état **Bound** et l'opération de restauration sera terminée. (BZ#2149654)

- Les VM créées à partir de modèles communs sur un cluster Single Node OpenShift (SNO) affichent une alerte **VMCannotBeEvicted** car la stratégie d'éviction par défaut du modèle est **LiveMigrate**. Vous pouvez ignorer cette alerte ou la supprimer en mettant à jour la stratégie d'éviction de la VM. (BZ#2092412)

- La désinstallation d'OpenShift Virtualization ne supprime pas les étiquettes de nœuds **feature.node.kubevirt.io** créées par OpenShift Virtualization. Vous devez supprimer les étiquettes manuellement. (CNV-22036)

- Certaines annotations de revendication de volume persistant (PVC) créées par Containerized Data Importer (CDI) peuvent entraîner un blocage indéfini de l'opération de restauration de l'instantané de la machine virtuelle. (BZ#2070366)
  - En guise de solution de contournement, vous pouvez supprimer les annotations manuellement :
    1. Obtenir le nom de la ressource personnalisée (CR) VirtualMachineSnapshotContent à partir de la valeur **status.virtualMachineSnapshotContentName** dans la CR **VirtualMachineSnapshot**.
    2. Modifiez le CR **VirtualMachineSnapshotContent** et supprimez toutes les lignes qui contiennent **k8s.io/cloneRequest**.
    3. Si vous n'avez pas spécifié de valeur pour **spec.dataVolumeTemplates** dans l'objet **VirtualMachine**, supprimez tous les objets **DataVolume** et **PersistentVolumeClaim** de cet espace de noms lorsque les deux conditions suivantes sont remplies :
       a. Le nom de l'objet commence par **restore-**.
       b. L'objet n'est pas référencé par les machines virtuelles.
       Cette étape est facultative si vous avez spécifié une valeur pour **spec.dataVolumeTemplates**.
    4. Répétez l'opération de restauration avec le CR **VirtualMachineSnapshot** mis à jour.

- Les machines virtuelles Windows 11 ne démarrent pas sur les clusters fonctionnant en mode FIPS. Windows 11 requiert par défaut un dispositif TPM (trusted platform module). Cependant, le paquetage **swtpm** (émulateur TPM logiciel) est incompatible avec FIPS. (BZ#2089301)

- Si votre cluster OpenShift Container Platform utilise OVN-Kubernetes comme fournisseur d'interface réseau de conteneurs (CNI) par défaut, vous ne pouvez pas attacher un pont Linux ou un périphérique de liaison à l'interface par défaut d'un hôte en raison d'un changement dans la topologie du réseau hôte d'OVN-Kubernetes. (BZ#1885605)

    - Comme solution de contournement, vous pouvez utiliser une interface réseau secondaire connectée à votre hôte, ou basculer vers le fournisseur CNI par défaut d'OpenShift SDN.

- Dans certains cas, plusieurs machines virtuelles peuvent monter le même PVC en mode lecture-écriture, ce qui peut entraîner une corruption des données. (BZ#1992753)

    - En guise de solution, évitez d'utiliser un seul PVC en mode lecture-écriture avec plusieurs machines virtuelles.

- Le Pod Disruption Budget (PDB) empêche les interruptions de pods pour les images de machines virtuelles migrables. Si le PDB détecte une interruption de pod, **openshift-monitoring** envoie une alerte **PodDisruptionBudgetAtLimit** toutes les 60 minutes pour les images de machines virtuelles qui utilisent la stratégie d'éviction **LiveMigrate**. (BZ#2026733)

    - Pour contourner le problème, il est possible de faire taire les alertes.

- OpenShift Virtualization lie un jeton de compte de service utilisé par un pod à ce pod spécifique. OpenShift Virtualization implémente un volume de compte de service en créant une image disque qui contient un jeton. Si vous migrez une VM, le volume de compte de service devient invalide. (BZ#2037611)

    - En guise de solution de contournement, utilisez des comptes d'utilisateur plutôt que des comptes de service, car les jetons de compte d'utilisateur ne sont pas liés à un module spécifique.

- Si vous clonez plus de 100 VM à l'aide de la stratégie de clonage **csi-clone**, le CSI Ceph risque de ne pas purger les clones. La suppression manuelle des clones peut également échouer. (BZ#2055595)

    - En guise de solution, vous pouvez redémarrer le site **ceph-mgr** pour purger les clones de VM.

CHAPITRE 6. INSTALLING

6.1. PRÉPARER VOTRE CLUSTER POUR OPENSHIFT VIRTUALIZATION

Consultez cette section avant d'installer OpenShift Virtualization pour vous assurer que votre cluster répond aux exigences.

IMPORTANT

Vous pouvez utiliser n'importe quelle méthode d'installation, y compris le provisionnement par l'utilisateur, le provisionnement par l'installateur ou l'installation assistée, pour déployer OpenShift Container Platform. Cependant, la méthode d'installation et la topologie du cluster peuvent affecter les fonctionnalités d'OpenShift Virtualization, telles que les snapshots ou la migration en direct.

Mode FIPS

Si vous installez votre cluster en mode FIPS, aucune configuration supplémentaire n'est requise pour OpenShift Virtualization.

6.1.1. Exigences en matière de matériel et de système d'exploitation

Passez en revue les exigences suivantes en matière de matériel et de système d'exploitation pour OpenShift Virtualization.

Plates-formes prises en charge

- Serveurs métalliques nus sur site
- Instances Amazon Web Services bare metal. Voir Déployer la virtualisation OpenShift sur des nœuds AWS Bare Metal pour plus de détails.
- Serveurs IBM Cloud Bare Metal. Voir Déployer la virtualisation OpenShift sur les nœuds IBM Cloud Bare Metal pour plus de détails.

IMPORTANT

L'installation d'OpenShift Virtualization sur des instances AWS bare metal ou sur des serveurs IBM Cloud bare metal est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de faire part de leurs commentaires au cours du processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

- **Bare metal instances or servers offered by other cloud providers are not supported.**

Exigences en matière de CPU

- Pris en charge par Red Hat Enterprise Linux (RHEL) 8
- Prise en charge des extensions de CPU Intel 64 ou AMD64
- Extensions de virtualisation matérielle Intel VT ou AMD-V activées
- Indicateur NX (pas d'exécution) activé

Storage requirements

- Pris en charge par OpenShift Container Platform

Operating system requirements

- Red Hat Enterprise Linux CoreOS (RHCOS) installé sur les nœuds de travail

NOTE

Les nœuds de travail RHEL ne sont pas pris en charge.

- Si votre cluster utilise des nœuds de travail dotés de différentes unités centrales, des échecs de migration en direct peuvent se produire car les unités centrales n'ont pas toutes les mêmes capacités. Pour éviter de tels échecs, utilisez des CPU ayant une capacité appropriée pour chaque nœud et définissez l'affinité de nœud sur vos machines virtuelles afin de garantir la réussite de la migration. Pour plus d'informations, voir Configuration d'une règle d'affinité de nœuds requise.

Ressources supplémentaires

- À propos de RHCOS.
- Catalogue de l'écosystème Red Hat pour les processeurs pris en charge.
- Stockage pris en charge .

6.1.2. Exigences en matière de frais généraux pour les ressources physiques

OpenShift Virtualization est une extension d'OpenShift Container Platform et impose des frais généraux supplémentaires que vous devez prendre en compte lors de la planification d'un cluster. Chaque machine de cluster doit répondre aux exigences suivantes en matière de frais généraux, en plus des exigences d'OpenShift Container Platform. La sursouscription des ressources physiques dans un cluster peut affecter les performances.

IMPORTANT

Les chiffres indiqués dans cette documentation sont basés sur la méthodologie de test et la configuration de Red Hat. Ces chiffres peuvent varier en fonction de votre propre configuration et de vos environnements.

6.1.2.1. Surcharge de mémoire

Calculez les valeurs de surcharge mémoire pour OpenShift Virtualization en utilisant les équations ci-dessous.

Surcharge de mémoire de la grappe

```
Memory overhead per infrastructure node ≈ 150 MiB
```

```
Memory overhead per worker node ≈ 360 MiB
```

En outre, les ressources de l'environnement de virtualisation OpenShift nécessitent un total de 2179 MiB de RAM qui est réparti sur tous les nœuds de l'infrastructure.

Surcharge de la mémoire de la machine virtuelle

```
Memory overhead per virtual machine ≈ (1.002 * requested memory) + 146 MiB  \
            + 8 MiB * (number of vCPUs) \ 1
          + 16 MiB * (number of graphics devices) 2
```

1 Nombre d'unités centrales virtuelles demandées par la machine virtuelle

2 Nombre de cartes graphiques virtuelles demandées par la machine virtuelle

Si votre environnement comprend un périphérique réseau SR-IOV (Single Root I/O Virtualization) ou une unité de traitement graphique (GPU), allouez 1 Go de mémoire supplémentaire pour chaque périphérique.

6.1.2.2. Frais généraux de l'unité centrale

Calculez les exigences de surcharge du processeur de cluster pour OpenShift Virtualization en utilisant l'équation ci-dessous. La surcharge de processeur par machine virtuelle dépend de votre configuration individuelle.

Surcharge de l'unité centrale de la grappe

```
CPU overhead for infrastructure nodes ≈ 4 cores
```

OpenShift Virtualization augmente l'utilisation globale des services de niveau cluster tels que la journalisation, le routage et la surveillance. Pour prendre en compte cette charge de travail, assurez-vous que les nœuds qui hébergent des composants d'infrastructure ont une capacité allouée pour 4 cœurs supplémentaires (4000 millicores) répartis sur ces nœuds.

```
CPU overhead for worker nodes ≈ 2 cores + CPU overhead per virtual machine
```

Chaque nœud de travailleur qui héberge des machines virtuelles doit avoir une capacité de 2 cœurs supplémentaires (2000 millicores) pour les charges de travail de gestion d'OpenShift Virtualization en plus des CPU requis pour les charges de travail des machines virtuelles.

Surcharge de l'unité centrale de la machine virtuelle

Si des CPU dédiés sont demandés, il y a un impact de 1:1 sur les besoins en CPU du cluster. Sinon, il n'y a pas de règles spécifiques concernant le nombre de CPU dont une machine virtuelle a besoin.

6.1.2.3. Stockage en hauteur

Utilisez les directives ci-dessous pour estimer les besoins en frais généraux de stockage pour votre environnement OpenShift Virtualization.

Frais généraux de stockage en grappe

> Aggregated storage overhead per node ≈ 10 GiB

10 GiB est l'impact estimé du stockage sur disque pour chaque nœud du cluster lorsque vous installez OpenShift Virtualization.

Surcharge de stockage de la machine virtuelle

Les frais généraux de stockage par machine virtuelle dépendent des demandes spécifiques d'allocation de ressources au sein de la machine virtuelle. La demande peut concerner le stockage éphémère sur le nœud ou les ressources de stockage hébergées ailleurs dans le cluster. OpenShift Virtualization n'alloue actuellement aucun stockage éphémère supplémentaire pour le conteneur en cours d'exécution lui-même.

6.1.2.4. Exemple :

En tant qu'administrateur de cluster, si vous prévoyez d'héberger 10 machines virtuelles dans le cluster, chacune avec 1 Go de RAM et 2 vCPU, l'impact de la mémoire sur le cluster est de 11,68 Go. L'impact estimé du stockage sur disque pour chaque nœud de la grappe est de 10 Go et l'impact du CPU pour les nœuds de travail qui hébergent les charges de travail des machines virtuelles est d'un minimum de 2 cœurs.

6.1.3. Maximums d'objets

Lors de la planification de votre cluster, vous devez tenir compte des maximums d'objets testés suivants :

- Maximums d'objets OpenShift Container Platform .
- Maximums d'objets de virtualisation OpenShift .

6.1.4. Environnements réseau restreints

Si vous installez OpenShift Virtualization dans un environnement restreint sans connectivité internet, vous devez configurer Operator Lifecycle Manager pour les réseaux restreints .

Si vous avez une connectivité internet limitée, vous pouvez configurer la prise en charge du proxy dans Operator Lifecycle Manager pour accéder à l'OperatorHub fourni par Red Hat.

6.1.5. Migration en direct

La migration en direct doit répondre aux exigences suivantes :

- Stockage partagé avec mode d'accès **ReadWriteMany** (RWX).
- Mémoire vive et bande passante suffisantes.
- Si la machine virtuelle utilise un modèle de CPU hôte, les nœuds doivent prendre en charge le modèle de CPU hôte de la machine virtuelle.

6.1.6. Instantanés et clonage

Voir les fonctionnalités de stockage d'OpenShift Virtualization pour les exigences en matière d'instantanés et de clonage.

6.1.7. Options de haute disponibilité du cluster

Vous pouvez configurer l'une des options de haute disponibilité (HA) suivantes pour votre cluster :

- La haute disponibilité automatique pour l'infrastructure fournie par l'installateur (IPI) est disponible en déployant des contrôles de santé des machines.

NOTE

Dans les clusters OpenShift Container Platform installés à l'aide d'une infrastructure fournie par l'installateur et avec MachineHealthCheck correctement configuré, si un nœud échoue le MachineHealthCheck et devient indisponible pour le cluster, il est recyclé. Ce qui se passe ensuite avec les machines virtuelles exécutées sur le nœud défaillant dépend d'une série de conditions. Voir À propos des stratégies d'exécution pour les machines virtuelles pour des informations plus détaillées sur les résultats potentiels et la façon dont les stratégies d'exécution affectent ces résultats.

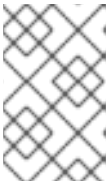
- La haute disponibilité automatique pour IPI et non-IPI est disponible en utilisant l'opérateur Node Health Check sur le cluster OpenShift Container Platform pour déployer le contrôleur **NodeHealthCheck**. Le contrôleur identifie les nœuds malsains et utilise l'opérateur Self Node Remediation pour remédier aux nœuds malsains.

IMPORTANT

Node Health Check Operator est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

- La haute disponibilité de n'importe quelle plate-forme est possible en utilisant un système de surveillance ou une personne qualifiée pour surveiller la disponibilité des nœuds. Lorsqu'un nœud est perdu, arrêtez-le et exécutez **oc delete node <lost_node>**.



NOTE

Sans un système de surveillance externe ou une personne qualifiée pour surveiller l'état des nœuds, les machines virtuelles perdent leur haute disponibilité.

6.2. SPÉCIFIER DES NŒUDS POUR LES COMPOSANTS D'OPENSHIFT VIRTUALIZATION

Spécifiez les nœuds où vous souhaitez déployer les opérateurs de virtualisation OpenShift, les charges de travail et les contrôleurs en configurant les règles de placement des nœuds.

NOTE

Vous pouvez configurer le placement des nœuds pour certains composants après avoir installé OpenShift Virtualization, mais il ne doit pas y avoir de machines virtuelles présentes si vous souhaitez configurer le placement des nœuds pour les charges de travail.

6.2.1. A propos de l'emplacement des nœuds pour les composants de virtualisation

Vous pourriez vouloir personnaliser l'endroit où OpenShift Virtualization déploie ses composants pour vous assurer que :

- Les machines virtuelles ne se déploient que sur les nœuds destinés aux charges de travail de virtualisation.
- Les opérateurs ne se déploient que sur les nœuds d'infrastructure.
- Certains nœuds ne sont pas affectés par OpenShift Virtualization. Par exemple, vous avez des charges de travail non liées à la virtualisation qui s'exécutent sur votre cluster, et vous voulez que ces charges de travail soient isolées d'OpenShift Virtualization.

6.2.1.1. Comment appliquer les règles de placement des nœuds aux composants de virtualisation ?

Vous pouvez spécifier les règles de placement des nœuds pour un composant en éditant l'objet correspondant directement ou en utilisant la console web.

- Pour les opérateurs de virtualisation OpenShift que Operator Lifecycle Manager (OLM) déploie, modifiez directement l'objet OLM **Subscription**. Actuellement, vous ne pouvez pas configurer les règles de placement des nœuds pour l'objet **Subscription** à l'aide de la console Web.
- Pour les composants que les opérateurs de virtualisation OpenShift déploient, modifiez directement l'objet **HyperConverged** ou configurez-le à l'aide de la console web lors de l'installation d'OpenShift Virtualization.
- Pour le provisionneur de chemins d'accès, modifiez l'objet **HostPathProvisioner** directement ou configurez-le à l'aide de la console Web.

AVERTISSEMENT

Vous devez planifier le hostpath provisioner et les composants de virtualisation sur les mêmes nœuds. Sinon, les pods de virtualisation qui utilisent le hostpath provisioner ne peuvent pas s'exécuter.

En fonction de l'objet, vous pouvez utiliser un ou plusieurs des types de règles suivants :

nodeSelector

Permet aux pods d'être planifiés sur des nœuds étiquetés avec la ou les paires clé-valeur que vous spécifiez dans ce champ. Le nœud doit avoir des étiquettes qui correspondent exactement à toutes les paires répertoriées.

affinity

Permet d'utiliser une syntaxe plus expressive pour définir des règles qui font correspondre des nœuds à des pods. Affinity permet également de nuancer la manière dont les règles sont appliquées. Par exemple, vous pouvez spécifier qu'une règle est une préférence plutôt qu'une exigence absolue, de sorte que les modules sont toujours programmés si la règle n'est pas respectée.

tolerations

Permet aux pods d'être planifiés sur des nœuds qui ont des taches correspondantes. Si une tare est appliquée à un nœud, ce nœud n'accepte que les pods qui tolèrent la tare.

6.2.1.2. Emplacement du nœud dans l'objet OLM Subscription

Pour spécifier les nœuds où OLM déploie les opérateurs de virtualisation OpenShift, modifiez l'objet **Subscription** pendant l'installation de la virtualisation OpenShift. Vous pouvez inclure des règles de placement des nœuds dans le champ **spec.config**, comme le montre l'exemple suivant :

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: hco-operatorhub
  namespace: openshift-cnv
spec:
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  name: kubevirt-hyperconverged
  startingCSV: kubevirt-hyperconverged-operator.v4.12.2
  channel: "stable"
  config: 1
```

1 Le champ **config** prend en charge **nodeSelector** et **tolerations**, mais pas **affinity**.

6.2.1.3. Placement des nœuds dans l'objet HyperConverged

Pour spécifier les nœuds où OpenShift Virtualization déploie ses composants, vous pouvez inclure l'objet **nodePlacement** dans le fichier de ressources personnalisées (CR) HyperConverged Cluster que vous créez lors de l'installation d'OpenShift Virtualization. Vous pouvez inclure **nodePlacement** dans les champs **spec.infra** et **spec.workloads**, comme le montre l'exemple suivant :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  infra:
    nodePlacement: 1
    ...
  workloads:
    nodePlacement:
    ...
```

1 Les champs **nodePlacement** supportent les champs **nodeSelector**, **affinity** et **tolerations**.

6.2.1.4. Placement du nœud dans l'objet HostPathProvisioner

Vous pouvez configurer les règles de placement des nœuds dans le champ **spec.workload** de l'objet **HostPathProvisioner** que vous créez lorsque vous installez le provisionneur de chemins d'accès.

```
apiVersion: hostpathprovisioner.kubevirt.io/v1beta1
kind: HostPathProvisioner
metadata:
  name: hostpath-provisioner
spec:
  imagePullPolicy: IfNotPresent
  pathConfig:
    path: "</path/to/backing/directory>"
    useNamingPrefix: false
  workload: 1
```

1 Le champ **workload** supporte les champs **nodeSelector**, **affinity** et **tolerations**.

6.2.1.5. Ressources supplémentaires

- Spécifier des nœuds pour les machines virtuelles
- Placer des pods sur des nœuds spécifiques en utilisant des sélecteurs de nœuds
- Contrôle du placement des pods sur les nœuds à l'aide de règles d'affinité des nœuds
- Contrôle du placement de pods à l'aide de taches de nœuds
- Installer OpenShift Virtualization à l'aide du CLI
- Installer OpenShift Virtualization à l'aide de la console web
- Configuration du stockage local pour les machines virtuelles

6.2.2. Exemples de manifestes

Les exemples de fichiers YAML suivants utilisent les objets **nodePlacement**, **affinity**, et **tolerations** pour personnaliser l'emplacement des nœuds pour les composants OpenShift Virtualization.

6.2.2.1. Objet d'abonnement du gestionnaire du cycle de vie de l'opérateur

6.2.2.1.1. Exemple : Placement d'un nœud avec nodeSelector dans l'objet OLM Subscription

Dans cet exemple, **nodeSelector** est configuré pour qu'OLM place les opérateurs de virtualisation OpenShift sur les nœuds étiquetés **example.io/example-infra-key = example-infra-value**.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: hco-operatorhub
```

```
  namespace: openshift-cnv
spec:
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  name: kubevirt-hyperconverged
  startingCSV: kubevirt-hyperconverged-operator.v4.12.2
  channel: "stable"
  config:
    nodeSelector:
      example.io/example-infra-key: example-infra-value
```

6.2.2.1.2. Exemple : Placement des nœuds avec tolérances dans l'objet OLM Abonnement

Dans cet exemple, les nœuds réservés à OLM pour déployer les opérateurs de virtualisation OpenShift sont étiquetés avec le taint **key=virtualization:NoSchedule**. Seuls les pods avec les tolérances correspondantes sont planifiés sur ces nœuds.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: hco-operatorhub
  namespace: openshift-cnv
spec:
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  name: kubevirt-hyperconverged
  startingCSV: kubevirt-hyperconverged-operator.v4.12.2
  channel: "stable"
  config:
    tolerations:
    - key: "key"
      operator: "Equal"
      value: "virtualization"
      effect: "NoSchedule"
```

6.2.2.2. Objet hyperconvergé

6.2.2.2.1. Exemple : Placement d'un nœud avec nodeSelector dans le cluster hyperconvergé CR

Dans cet exemple, **nodeSelector** est configuré de manière à ce que les ressources d'infrastructure soient placées sur les nœuds identifiés par **example.io/example-infra-key = example-infra-value** et que les charges de travail soient placées sur les nœuds identifiés par **example.io/example-workloads-key = example-workloads-value**.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  infra:
    nodePlacement:
      nodeSelector:
        example.io/example-infra-key: example-infra-value
```

```
  workloads:
    nodePlacement:
      nodeSelector:
        example.io/example-workloads-key: example-workloads-value
```

6.2.2.2.2. Exemple : Placement de nœuds avec affinité dans le cluster hyperconvergé CR

Dans cet exemple, **affinity** est configuré de manière à ce que les ressources d'infrastructure soient placées sur les nœuds identifiés par **example.io/example-infra-key = example-value** et que les charges de travail soient placées sur les nœuds identifiés par **example.io/example-workloads-key = example-workloads-value**. Les nœuds disposant de plus de huit CPU sont privilégiés pour les charges de travail, mais s'ils ne sont pas disponibles, les pods sont tout de même planifiés.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  infra:
    nodePlacement:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: example.io/example-infra-key
                operator: In
                values:
                - example-infra-value
  workloads:
    nodePlacement:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: example.io/example-workloads-key
                operator: In
                values:
                - example-workloads-value
          preferredDuringSchedulingIgnoredDuringExecution:
          - weight: 1
            preference:
              matchExpressions:
              - key: example.io/num-cpus
                operator: Gt
                values:
                - 8
```

6.2.2.2.3. Exemple : Placement de nœuds avec tolérances dans le cluster hyperconvergé CR

Dans cet exemple, les nœuds réservés aux composants OpenShift Virtualization sont étiquetés avec le taint **key=virtualization:NoSchedule**. Seuls les pods avec les tolérances correspondantes sont planifiés sur ces nœuds.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  workloads:
    nodePlacement:
      tolerations:
      - key: "key"
        operator: "Equal"
        value: "virtualization"
        effect: "NoSchedule"
```

6.2.2.3. Objet HostPathProvisioner

6.2.2.3.1. Exemple : Placement d'un nœud avec nodeSelector dans l'objet HostPathProvisioner

Dans cet exemple, **nodeSelector** est configuré de manière à ce que les charges de travail soient placées sur les nœuds identifiés par **example.io/example-workloads-key = example-workloads-value**.

```
apiVersion: hostpathprovisioner.kubevirt.io/v1beta1
kind: HostPathProvisioner
metadata:
  name: hostpath-provisioner
spec:
  imagePullPolicy: IfNotPresent
  pathConfig:
    path: "</path/to/backing/directory>"
    useNamingPrefix: false
  workload:
    nodeSelector:
      example.io/example-workloads-key: example-workloads-value
```

6.3. INSTALLER OPENSHIFT VIRTUALIZATION À L'AIDE DE LA CONSOLE WEB

Installez OpenShift Virtualization pour ajouter des fonctionnalités de virtualisation à votre cluster OpenShift Container Platform.

Vous pouvez utiliser la console web d' OpenShift Container Platform 4.12 pour vous abonner et déployer les opérateurs de virtualisation OpenShift.

6.3.1. Installation de l'opérateur de virtualisation OpenShift

Vous pouvez installer OpenShift Virtualization Operator depuis la console web d'OpenShift Container Platform.

Conditions préalables

- Installez OpenShift Container Platform 4.12 sur votre cluster.

- Connectez-vous à la console web de OpenShift Container Platform en tant qu'utilisateur avec les permissions **cluster-admin**.

Procédure

1. Dans la perspective **Administrator**, cliquez sur **Operators → OperatorHub**.
2. Dans le champ **Filter by keyword**, tapez **OpenShift Virtualization**.
3. Sélectionnez la tuile **OpenShift Virtualization**.
4. Lisez les informations sur l'opérateur et cliquez sur **Install**.
5. Sur la page **Install Operator**:
   a. Sélectionnez **stable** dans la liste des options **Update Channel** disponibles. Cela garantit que vous installez la version d'OpenShift Virtualization qui est compatible avec votre version d'OpenShift Container Platform.
   b. Pour **Installed Namespace**, assurez-vous que l'option **Operator recommended namespace** est sélectionnée. Cela installe l'opérateur dans l'espace de noms obligatoire **openshift-cnv**, qui est automatiquement créé s'il n'existe pas.

AVERTISSEMENT

La tentative d'installation de l'opérateur de virtualisation OpenShift dans un espace de noms autre que **openshift-cnv** entraîne l'échec de l'installation.

   c. Pour **Approval Strategy**, il est fortement recommandé de sélectionner **Automatic**, qui est la valeur par défaut, afin qu'OpenShift Virtualization se mette automatiquement à jour lorsqu'une nouvelle version est disponible dans le canal de mise à jour **stable**.
   Bien qu'il soit possible de sélectionner la stratégie d'approbation **Manual**, cela est déconseillé en raison du risque élevé qu'elle présente pour le support et la fonctionnalité de votre cluster. Ne choisissez **Manual** que si vous comprenez parfaitement ces risques et que vous ne pouvez pas utiliser **Automatic**.

AVERTISSEMENT

Comme OpenShift Virtualization n'est pris en charge que lorsqu'il est utilisé avec la version correspondante d'OpenShift Container Platform, les mises à jour manquantes d'OpenShift Virtualization peuvent entraîner l'absence de prise en charge de votre cluster.

6. Cliquez sur **Install** pour mettre l'opérateur à la disposition de l'espace de noms **openshift-cnv**.

7. Lorsque l'installation de l'opérateur a réussi, cliquez sur **Create HyperConverged**.

8. Facultatif : Configurez les options de placement des nœuds **Infra** et **Workloads** pour les composants OpenShift Virtualization.

9. Cliquez sur **Create** pour lancer OpenShift Virtualization.

Vérification

- Naviguez vers la page **Workloads → Pods** et surveillez les pods de virtualisation OpenShift jusqu'à ce qu'ils soient tous **Running**. Une fois que tous les pods affichent l'état **Running**, vous pouvez utiliser OpenShift Virtualization.

6.3.2. Prochaines étapes

Il est possible que vous souhaitiez configurer en plus les composants suivants :

- Le hostpath provisioner est un provisionneur de stockage local conçu pour OpenShift Virtualization. Si vous souhaitez configurer le stockage local pour les machines virtuelles, vous devez d'abord activer le hostpath provisioner.

6.4. INSTALLER OPENSHIFT VIRTUALIZATION À L'AIDE DU CLI

Installez OpenShift Virtualization pour ajouter des fonctionnalités de virtualisation à votre cluster OpenShift Container Platform. Vous pouvez souscrire et déployer les opérateurs de virtualisation OpenShift en utilisant la ligne de commande pour appliquer les manifestes à votre cluster.

NOTE

Pour spécifier les nœuds sur lesquels OpenShift Virtualization doit installer ses composants, configurez les règles de placement des nœuds .

6.4.1. Conditions préalables

- Installez OpenShift Container Platform 4.12 sur votre cluster.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous en tant qu'utilisateur disposant des privilèges **cluster-admin**.

6.4.2. S'abonner au catalogue OpenShift Virtualization en utilisant le CLI

Avant d'installer OpenShift Virtualization, vous devez vous abonner au catalogue OpenShift Virtualization. L'abonnement permet à l'espace de noms **openshift-cnv** d'accéder aux opérateurs de virtualisation OpenShift.

Pour vous abonner, configurez les objets **Namespace**, **OperatorGroup** et **Subscription** en appliquant un manifeste unique à votre cluster.

Procédure

1. Créez un fichier YAML contenant le manifeste suivant :

```
apiVersion: v1
```

```
kind: Namespace
metadata:
  name: openshift-cnv
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: kubevirt-hyperconverged-group
  namespace: openshift-cnv
spec:
  targetNamespaces:
    - openshift-cnv
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: hco-operatorhub
  namespace: openshift-cnv
spec:
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  name: kubevirt-hyperconverged
  startingCSV: kubevirt-hyperconverged-operator.v4.12.2
  channel: "stable" (1)
```

(1) L'utilisation du canal **stable** garantit que vous installez la version d'OpenShift Virtualization compatible avec votre version d'OpenShift Container Platform.

2. Créez les objets **Namespace**, **OperatorGroup**, et **Subscription** requis pour OpenShift Virtualization en exécutant la commande suivante :

```
$ oc apply -f <file name>.yaml
```

NOTE

Vous pouvez configurer les paramètres de rotation des certificats dans le fichier YAML.

6.4.3. Déployer l'opérateur de virtualisation OpenShift en utilisant le CLI

Vous pouvez déployer l'opérateur de virtualisation OpenShift en utilisant le CLI **oc**.

Conditions préalables

- Un abonnement actif au catalogue OpenShift Virtualization dans l'espace de noms **openshift-cnv**.

Procédure

1. Créez un fichier YAML contenant le manifeste suivant :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
```

```
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
```

2. Déployez l'opérateur de virtualisation OpenShift en exécutant la commande suivante :

```
oc apply -f <nom_du_fichier>.yaml
```

Vérification

- Assurez-vous qu'OpenShift Virtualization s'est déployé avec succès en observant le site **PHASE** de la version du service de cluster (CSV) dans l'espace de noms **openshift-cnv**. Exécutez la commande suivante :

```
$ watch oc get csv -n openshift-cnv
```

La sortie suivante s'affiche si le déploiement a réussi :

Exemple de sortie

```
NAME                                      DISPLAY                    VERSION   REPLACES   PHASE
kubevirt-hyperconverged-operator.v4.12.2  OpenShift Virtualization   4.12.2
Succeeded
```

6.4.4. Prochaines étapes

Il est possible que vous souhaitiez configurer en plus les composants suivants :

- Le hostpath provisioner est un provisionneur de stockage local conçu pour OpenShift Virtualization. Si vous souhaitez configurer le stockage local pour les machines virtuelles, vous devez d'abord activer le hostpath provisioner.

6.5. INSTALLATION DU CLIENT VIRTCTL

Le client **virtctl** est un utilitaire de ligne de commande pour gérer les ressources de virtualisation OpenShift. Il est disponible pour Linux, Windows et macOS.

6.5.1. Installation du client virtctl sous Linux, Windows et macOS

Téléchargez et installez le client **virtctl** pour votre système d'exploitation.

Procédure

1. Naviguez vers **Virtualization > Overview** dans la console web de OpenShift Container Platform.
2. Cliquez sur le lien **Download virtctl** dans le coin supérieur droit de la page et téléchargez le client **virtctl** pour votre système d'exploitation.
3. Installer **virtctl**:
   - Pour Linux :
     a. Décompresser le fichier d'archive :

```
$ tar -xvf <virtctl-version-distribution.arch>.tar.gz
```

b. Exécutez la commande suivante pour rendre le binaire **virtctl** exécutable :

```
$ chmod x <path/virtctl-file-name>
```

c. Déplacez le binaire **virtctl** dans un répertoire de votre variable d'environnement **PATH**. Vous pouvez vérifier votre chemin d'accès en exécutant la commande suivante :

```
$ echo $PATH
```

d. Définir la variable d'environnement **KUBECONFIG**:

```
$ export KUBECONFIG=/home/<user>/clusters/current/auth/kubeconfig
```

- Pour Windows :

a. Décompresser le fichier d'archive.

b. Naviguez dans la hiérarchie des dossiers extraits et double-cliquez sur le fichier exécutable **virtctl** pour installer le client.

c. Déplacez le binaire **virtctl** dans un répertoire de votre variable d'environnement **PATH**. Vous pouvez vérifier votre chemin d'accès en exécutant la commande suivante :

```
C:\N> path
```

- Pour macOS :

a. Décompresser le fichier d'archive.

b. Déplacez le binaire **virtctl** dans un répertoire de votre variable d'environnement **PATH**. Vous pouvez vérifier votre chemin d'accès en exécutant la commande suivante :

```
echo $PATH
```

6.5.2. Installer virtctl en tant que RPM

Vous pouvez installer le client **virtctl** sur Red Hat Enterprise Linux (RHEL) en tant que RPM après avoir activé le dépôt OpenShift Virtualization.

6.5.2.1. Activation des référentiels OpenShift Virtualization

Activez le dépôt OpenShift Virtualization pour votre version de Red Hat Enterprise Linux (RHEL).

Conditions préalables

- Votre système est enregistré sur un compte Red Hat avec un abonnement actif au droit "Red Hat Container Native Virtualization".

Procédure

- Activez le référentiel OpenShift Virtualization approprié pour votre système d'exploitation en utilisant l'outil CLI **subscription-manager**.
  - Pour activer le référentiel pour RHEL 8, exécutez :

```
# subscription-manager repos --enable cnv-4.12-for-rhel-8-x86_64-rpms
```

  - Pour activer le référentiel pour RHEL 7, exécutez :

```
# subscription-manager repos --enable rhel-7-server-cnv-4.12-rpms
```

6.5.2.2. Installation du client virtctl à l'aide de l'utilitaire yum

Installez le client **virtctl** à partir du paquetage **kubevirt-virtctl**.

Conditions préalables

- Vous avez activé un référentiel OpenShift Virtualization sur votre système Red Hat Enterprise Linux (RHEL).

Procédure

- Installez le paquetage **kubevirt-virtctl**:

```
# yum install kubevirt-virtctl
```

6.5.3. Ressources supplémentaires

- Utiliser les outils CLI pour OpenShift Virtualization.

6.6. DÉSINSTALLATION D'OPENSHIFT VIRTUALIZATION

Vous désinstallez OpenShift Virtualization en utilisant la console web ou l'interface de ligne de commande (CLI) pour supprimer les charges de travail OpenShift Virtualization, l'opérateur et ses ressources.

6.6.1. Désinstallation d'OpenShift Virtualization à l'aide de la console web

Vous désinstallez OpenShift Virtualization en utilisant la console web pour effectuer les tâches suivantes :

1. Supprimer le CR **HyperConverged** .
2. Supprimer l'opérateur de virtualisation OpenShift .
3. Supprimer l'espace de noms **openshift-cnv** .
4. Supprimez les définitions de ressources personnalisées (CRD) d'OpenShift Virtualization .

IMPORTANT

Vous devez d'abord supprimer toutes les machines virtuelles et les instances de machines virtuelles.

Vous ne pouvez pas désinstaller OpenShift Virtualization tant que ses charges de travail restent sur le cluster.

6.6.1.1. Suppression de la ressource personnalisée HyperConverged

Pour désinstaller OpenShift Virtualization, vous devez d'abord supprimer la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform en utilisant un compte avec des permissions **cluster-admin**.

Procédure

1. Naviguez jusqu'à la page **Operators → Installed Operators**.
2. Sélectionnez l'opérateur de virtualisation OpenShift.
3. Cliquez sur l'onglet **OpenShift Virtualization Deployment**.
4. Cliquez sur le menu Options  à côté de **kubevirt-hyperconverged** et sélectionnez **Delete HyperConverged**.
5. Cliquez sur **Delete** dans la fenêtre de confirmation.

6.6.1.2. Suppression d'opérateurs d'une grappe à l'aide de la console web

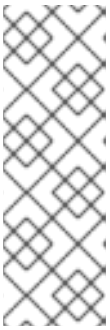
Les administrateurs de cluster peuvent supprimer les opérateurs installés dans un espace de noms sélectionné à l'aide de la console web.

Conditions préalables

- Vous avez accès à la console web d'un cluster OpenShift Container Platform en utilisant un compte avec les permissions **cluster-admin**.

Procédure

1. Naviguez jusqu'à la page **Operators → Installed Operators**.
2. Faites défiler ou saisissez un mot-clé dans le champ **Filter by name** pour trouver l'opérateur que vous souhaitez supprimer. Cliquez ensuite dessus.
3. Sur le côté droit de la page **Operator Details**, sélectionnez **Uninstall Operator** dans la liste **Actions**.
   Une boîte de dialogue **Uninstall Operator?** s'affiche.
4. Sélectionnez **Uninstall** pour supprimer l'opérateur, les déploiements de l'opérateur et les pods. Suite à cette action, l'opérateur cesse de fonctionner et ne reçoit plus de mises à jour.



NOTE

Cette action ne supprime pas les ressources gérées par l'opérateur, y compris les définitions de ressources personnalisées (CRD) et les ressources personnalisées (CR). Les tableaux de bord et les éléments de navigation activés par la console Web et les ressources hors cluster qui continuent de fonctionner peuvent nécessiter un nettoyage manuel. Pour les supprimer après la désinstallation de l'opérateur, vous devrez peut-être supprimer manuellement les CRD de l'opérateur.

6.6.1.3. Suppression d'un espace de noms à l'aide de la console web

Vous pouvez supprimer un espace de noms en utilisant la console web d'OpenShift Container Platform.

Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform en utilisant un compte avec des permissions **cluster-admin**.

Procédure

1. Naviguez jusqu'à **Administration → Namespaces**.
2. Localisez l'espace de noms que vous souhaitez supprimer dans la liste des espaces de noms.
3. À l'extrême droite de la liste des espaces de noms, sélectionnez **Delete Namespace** dans le menu Options .
4. Lorsque le volet **Delete Namespace** s'ouvre, saisissez le nom de l'espace de noms que vous souhaitez supprimer dans le champ.
5. Cliquez sur **Delete**.

6.6.1.4. Suppression des définitions de ressources personnalisées d'OpenShift Virtualization

Vous pouvez supprimer les définitions de ressources personnalisées (CRD) d'OpenShift Virtualization en utilisant la console web.

Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform en utilisant un compte avec des permissions **cluster-admin**.

Procédure

1. Naviguez jusqu'à **Administration → CustomResourceDefinitions**.
2. Sélectionnez le filtre **Label** et entrez **operators.coreos.com/kubevirt-hyperconverged.openshift-cnv** dans le champ **Search** pour afficher les CRD de virtualisation OpenShift.

3. Cliquez sur le menu Options  à côté de chaque CRD et sélectionnez **Delete CustomResourceDefinition**.

6.6.2. Désinstaller OpenShift Virtualization en utilisant le CLI

Vous pouvez désinstaller OpenShift Virtualization en utilisant le CLI OpenShift (**oc**).

Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform en utilisant un compte avec des permissions **cluster-admin**.
- Vous avez installé l'OpenShift CLI (**oc**).
- Vous avez supprimé toutes les machines virtuelles et les instances de machines virtuelles. Vous ne pouvez pas désinstaller OpenShift Virtualization alors que ses charges de travail restent sur le cluster.

Procédure

1. Supprimer la ressource personnalisée **HyperConverged**:

```
$ oc delete HyperConverged kubevirt-hyperconverged -n openshift-cnv
```

2. Supprimer l'abonnement à OpenShift Virtualization Operator :

```
$ oc delete subscription kubevirt-hyperconverged -n openshift-cnv
```

3. Supprimer la ressource OpenShift Virtualization **ClusterServiceVersion**:

```
$ oc delete csv -n openshift-cnv -l operators.coreos.com/kubevirt-hyperconverged.openshift-cnv
```

4. Listez les définitions de ressources personnalisées (CRD) d'OpenShift Virtualization en exécutant la commande **oc delete crd** avec l'option **dry-run**:

```
$ oc delete crd --dry-run=client -l operators.coreos.com/kubevirt-hyperconverged.openshift-cnv
```

Exemple de sortie

```
customresourcedefinition.apiextensions.k8s.io "cdis.cdi.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "hostpathprovisioners.hostpathprovisioner.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "hyperconvergeds.hco.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "kubevirts.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "networkaddonsconfigs.networkaddonsoperator.network.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "ssps.ssp.kubevirt.io" deleted (dry run)
customresourcedefinition.apiextensions.k8s.io "tektontasks.tektontasks.kubevirt.io" deleted (dry run)
```

–

5. Supprimez les CRD en exécutant la commande **oc delete crd** sans l'option **dry-run**:

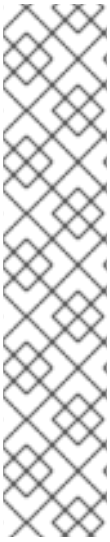
```
$ oc delete crd -l operators.coreos.com/kubevirt-hyperconverged.openshift-cnv
```

Ressources supplémentaires

- Suppression des machines virtuelles
- Suppression d'instances de machines virtuelles

CHAPITRE 7. MISE À JOUR DE LA VIRTUALISATION OPENSHIFT

Découvrez comment Operator Lifecycle Manager (OLM) fournit des mises à jour de flux z et de versions mineures pour OpenShift Virtualization.



NOTE

- Le Node Maintenance Operator (NMO) n'est plus livré avec OpenShift Virtualization. Vous pouvez installer le NMO à partir de **OperatorHub** dans la console web d'OpenShift Container Platform, ou en utilisant la CLI d'OpenShift (**oc**).
  Vous devez effectuer l'une des tâches suivantes avant de mettre à jour OpenShift Virtualization 4.11 à partir d'OpenShift Virtualization 4.10.2 et des versions ultérieures :
  - Sortir tous les nœuds du mode maintenance.
  - Installez l'ONM autonome et remplacez la ressource personnalisée (CR) **nodemaintenances.nodemaintenance.kubevirt.io** par une CR **nodemaintenances.nodemaintenance.medik8s.io**.

7.1. À PROPOS DE LA MISE À JOUR DE LA VIRTUALISATION OPENSHIFT

- Operator Lifecycle Manager (OLM) gère le cycle de vie de l'opérateur de virtualisation OpenShift. L'opérateur Marketplace, qui est déployé lors de l'installation d'OpenShift Container Platform, met les opérateurs externes à la disposition de votre cluster.
- OLM fournit un flux z et des mises à jour de versions mineures pour OpenShift Virtualization. Les mises à jour des versions mineures sont disponibles lorsque vous mettez à jour OpenShift Container Platform vers la version mineure suivante. Vous ne pouvez pas mettre à jour OpenShift Virtualization vers la version mineure suivante sans d'abord mettre à jour OpenShift Container Platform.
- Les abonnements à OpenShift Virtualization utilisent un canal de mise à jour unique nommé **stable**. Le canal **stable** assure la compatibilité des versions d'OpenShift Virtualization et d'OpenShift Container Platform.
- Si la stratégie d'approbation de votre abonnement est définie sur **Automatic**, le processus de mise à jour démarre dès qu'une nouvelle version de l'opérateur est disponible dans le canal **stable**. Il est fortement recommandé d'utiliser la stratégie d'approbation **Automatic** pour maintenir un environnement supportable. Chaque version mineure d'OpenShift Virtualization n'est prise en charge que si vous exécutez la version correspondante d'OpenShift Container Platform. Par exemple, vous devez exécuter OpenShift Virtualization 4.12 sur OpenShift Container Platform 4.12.
  - Bien qu'il soit possible de sélectionner la stratégie d'approbation **Manual**, cela n'est pas recommandé car cela risque de compromettre le support et la fonctionnalité de votre cluster. Avec la stratégie d'approbation **Manual**, vous devez approuver manuellement chaque mise à jour en attente. Si les mises à jour d'OpenShift Container Platform et d'OpenShift Virtualization ne sont pas synchronisées, votre cluster ne sera plus supporté.

- Le délai d'exécution d'une mise à jour dépend de votre connexion réseau. La plupart des mises à jour automatiques s'effectuent en quinze minutes.
- La mise à jour d'OpenShift Virtualization n'interrompt pas les connexions réseau.
- Les volumes de données et les revendications de volumes persistants qui leur sont associées sont préservés lors de la mise à jour.

IMPORTANT

Si vous avez des machines virtuelles en cours d'exécution qui utilisent le stockage hostpath provisioner, elles ne peuvent pas être migrées en direct et peuvent bloquer une mise à jour du cluster OpenShift Container Platform.

En guise de solution, vous pouvez reconfigurer les machines virtuelles de manière à ce qu'elles puissent être mises hors tension automatiquement lors d'une mise à jour du cluster. Supprimez le champ **evictionStrategy: LiveMigrate** et définissez le champ **runStrategy** sur **Always**.

7.1.1. A propos des mises à jour de la charge de travail

Lorsque vous mettez à jour OpenShift Virtualization, les charges de travail des machines virtuelles, y compris **libvirt**, **virt-launcher**, et **qemu**, se mettent à jour automatiquement si elles prennent en charge la migration en direct.

NOTE

Chaque machine virtuelle possède un pod **virt-launcher** qui exécute l'instance de machine virtuelle (VMI). Le pod **virt-launcher** exécute une instance de **libvirt**, qui est utilisée pour gérer le processus de la machine virtuelle (VM).

Vous pouvez configurer la manière dont les charges de travail sont mises à jour en modifiant la strophe **spec.workloadUpdateStrategy** de la ressource personnalisée (CR) **HyperConverged**. Il existe deux méthodes de mise à jour des charges de travail : **LiveMigrate** et **Evict**.

Comme la méthode **Evict** arrête les pods VMI, seule la stratégie de mise à jour **LiveMigrate** est activée par défaut.

Lorsque **LiveMigrate** est la seule stratégie de mise à jour activée :

- Les IMV qui prennent en charge la migration en direct sont migrées pendant le processus de mise à jour. L'invité de la VM se déplace dans un nouveau pod avec les composants mis à jour activés.
- Les IMV qui ne prennent pas en charge la migration en direct ne sont pas interrompues ni mises à jour.
    - Si un IMV dispose de la stratégie d'éviction **LiveMigrate** mais ne prend pas en charge la migration en direct, il n'est pas mis à jour.

Si vous activez à la fois **LiveMigrate** et **Evict**:

- Les IMV qui prennent en charge la migration en direct utilisent la stratégie de mise à jour **LiveMigrate**.

- Les IMV qui ne prennent pas en charge la migration en direct utilisent la stratégie de mise à jour **Evict**. Si une IMV est contrôlée par un objet **VirtualMachine** dont la valeur **runStrategy** est **always**, une nouvelle IMV est créée dans un nouveau module avec des composants mis à jour.

Tentatives de migration et délais d'attente
Lors de la mise à jour des charges de travail, la migration en direct échoue si un pod est dans l'état **Pending** pendant les périodes suivantes :

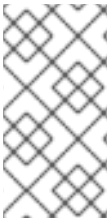
5 minutes

Si le pod est en attente parce qu'il est **Unschedulable**.

15 minutes

Si le pod est bloqué dans l'état d'attente pour une raison quelconque.

Lorsqu'une IMV ne parvient pas à migrer, le site **virt-controller** essaie de la migrer à nouveau. Il répète ce processus jusqu'à ce que toutes les IMV migrables soient exécutées sur les nouveaux pods **virt-launcher**. Cependant, si une IMV est mal configurée, ces tentatives peuvent se répéter indéfiniment.



NOTE

Chaque tentative correspond à un objet de migration. Seules les cinq tentatives les plus récentes sont conservées dans une mémoire tampon. Cela permet d'éviter que les objets de migration ne s'accumulent sur le système tout en conservant des informations pour le débogage.

7.1.2. À propos des mises à jour EUS-to-EUS

Chaque version mineure paire d'OpenShift Container Platform, y compris les versions 4.10 et 4.12, est une version EUS (Extended Update Support). Cependant, comme la conception de Kubernetes impose des mises à jour en série des versions mineures, vous ne pouvez pas passer directement d'une version EUS à la suivante.

Après avoir mis à jour la version EUS source vers la prochaine version mineure impaire, vous devez séquentiellement mettre à jour OpenShift Virtualization vers toutes les versions z-stream de cette version mineure qui se trouvent sur votre chemin de mise à jour. Lorsque vous avez mis à jour vers la dernière version z-stream applicable, vous pouvez alors mettre à jour OpenShift Container Platform vers la version mineure EUS cible.

Lorsque la mise à jour d'OpenShift Container Platform réussit, la mise à jour correspondante pour OpenShift Virtualization devient disponible. Vous pouvez maintenant mettre à jour OpenShift Virtualization vers la version EUS cible.

7.1.2.1. Préparation de la mise à jour

Avant de commencer une mise à jour EUS-to-EUS, vous devez :

- Mettez en pause les pools de configuration des machines des nœuds de travail avant de lancer une mise à jour EUS vers EUS afin que les travailleurs ne soient pas redémarrés deux fois.

- Désactivez les mises à jour automatiques de la charge de travail avant de commencer le processus de mise à jour. Cela permet d'éviter qu'OpenShift Virtualization ne migre ou n'expulse vos machines virtuelles (VM) jusqu'à ce que vous mettiez à jour votre version cible d'EUS.

NOTE

Par défaut, OpenShift Virtualization met automatiquement à jour les workloads, tels que le pod **virt-launcher**, lorsque vous mettez à jour l'OpenShift Virtualization Operator. Vous pouvez configurer ce comportement dans la strophe **spec.workloadUpdateStrategy** de la ressource personnalisée **HyperConverged**.

En savoir plus sur la préparation d'une mise à jour EUS-to-EUS .

7.2. PRÉVENTION DES MISES À JOUR DE LA CHARGE DE TRAVAIL LORS D'UNE MISE À JOUR EUS-TO-EUS

Lorsque vous passez d'une version Extended Update Support (EUS) à la suivante, vous devez désactiver manuellement les mises à jour automatiques des charges de travail pour empêcher OpenShift Virtualization de migrer ou d'expulser des charges de travail pendant le processus de mise à jour.

Conditions préalables

- Vous utilisez une version EUS d'OpenShift Container Platform et souhaitez passer à la version EUS suivante. Vous n'avez pas encore mis à jour la version impaire entre les deux.
- Vous avez lu "Preparing to perform an EUS-to-EUS update" (Préparation à la mise à jour EUS-to-EUS) et vous avez pris connaissance des mises en garde et des exigences relatives à votre cluster OpenShift Container Platform.
- Vous avez pausé les pools de configuration des machines des nœuds de travail comme indiqué dans la documentation d'OpenShift Container Platform.
- Il est recommandé d'utiliser la stratégie d'approbation par défaut **Automatic**. Si vous utilisez la stratégie d'approbation **Manual**, vous devez approuver toutes les mises à jour en attente dans la console web. Pour plus de détails, reportez-vous à la section "Approbation manuelle d'une mise à jour de l'opérateur en attente".

Procédure

1. Sauvegardez la configuration actuelle de **workloadUpdateMethods** en exécutant la commande suivante :

```
$ WORKLOAD_UPDATE_METHODS=$(oc get kv kubevirt-kubevirt-hyperconverged -n openshift-cnv -o jsonpath='{.spec.workloadUpdateStrategy.workloadUpdateMethods}')
```

2. Désactivez toutes les méthodes de mise à jour de la charge de travail en exécutant la commande suivante :

```
$ oc patch hco kubevirt-hyperconverged -n openshift-cnv --type json -p '[{"op":"replace","path":"/spec/workloadUpdateStrategy/workloadUpdateMethods", "value":[]}]'
```

Exemple de sortie

```
hyperconverged.hco.kubevirt.io/kubevirt-hyperconverged patched
```

3. Assurez-vous que l'opérateur **HyperConverged** est bien **Upgradeable** avant de continuer. Entrez la commande suivante et surveillez la sortie :

```
$ oc get hco kubevirt-hyperconverged -n openshift-cnv -o json | jq ".status.conditions"
```

Exemple 7.1. Exemple de sortie

```
[
  {
    "lastTransitionTime": "2022-12-09T16:29:11Z",
    "message": "Reconcile completed successfully",
    "observedGeneration": 3,
    "reason": "ReconcileCompleted",
    "status": "True",
    "type": "ReconcileComplete"
  },
  {
    "lastTransitionTime": "2022-12-09T20:30:10Z",
    "message": "Reconcile completed successfully",
    "observedGeneration": 3,
    "reason": "ReconcileCompleted",
    "status": "True",
    "type": "Available"
  },
  {
    "lastTransitionTime": "2022-12-09T20:30:10Z",
    "message": "Reconcile completed successfully",
    "observedGeneration": 3,
    "reason": "ReconcileCompleted",
    "status": "False",
    "type": "Progressing"
  },
  {
    "lastTransitionTime": "2022-12-09T16:39:11Z",
    "message": "Reconcile completed successfully",
    "observedGeneration": 3,
    "reason": "ReconcileCompleted",
    "status": "False",
    "type": "Degraded"
  },
  {
    "lastTransitionTime": "2022-12-09T20:30:10Z",
    "message": "Reconcile completed successfully",
    "observedGeneration": 3,
    "reason": "ReconcileCompleted",
    "status": "True",
    "type": "Upgradeable" (1)
  }
]
```

(1) L'opérateur de virtualisation OpenShift a le statut **Upgradeable**.

4. Mettez manuellement à jour votre cluster à partir de la version EUS source vers la version mineure suivante d'OpenShift Container Platform :

```
$ oc adm upgrade
```

Vérification

- Vérifiez la version actuelle en exécutant la commande suivante :

```
$ oc get clusterversion
```

NOTE

La mise à jour d'OpenShift Container Platform vers la version suivante est une condition préalable à la mise à jour d'OpenShift Virtualization. Pour plus de détails, reportez-vous à la section " Mise à jour des clusters " de la documentation d'OpenShift Container Platform.

5. Mettre à jour OpenShift Virtualization.
   - Avec la stratégie d'approbation par défaut **Automatic**, OpenShift Virtualization se met automatiquement à jour vers la version correspondante après la mise à jour d'OpenShift Container Platform.
   - Si vous utilisez la stratégie d'approbation **Manual**, approuvez les mises à jour en attente en utilisant la console web.
6. Surveillez la mise à jour d'OpenShift Virtualization en exécutant la commande suivante :

```
$ oc get csv -n openshift-cnv
```

7. Mettez à jour OpenShift Virtualization à chaque version de z-stream disponible pour la version mineure non-EUS, en surveillant chaque mise à jour en exécutant la commande montrée dans l'étape précédente.
8. Confirmez qu'OpenShift Virtualization a bien été mis à jour vers la dernière version z-stream de la version non-EUS en exécutant la commande suivante :

```
$ oc get hco kubevirt-hyperconverged -n openshift-cnv -o json | jq ".status.versions"
```

Exemple de sortie

```
[
  {
    "name": "operator",
    "version": "4.12.2"
  }
]
```

9. Attendez que l'opérateur **HyperConverged** ait l'état **Upgradeable** avant de procéder à la mise à jour suivante. Entrez la commande suivante et surveillez la sortie :

```
$ oc get hco kubevirt-hyperconverged -n openshift-cnv -o json | jq ".status.conditions"
```

10. Mettre à jour OpenShift Container Platform à la version EUS cible.

11. Confirmez que la mise à jour a réussi en vérifiant la version du cluster :

```
$ oc get clusterversion
```

12. Mettre à jour OpenShift Virtualization vers la version EUS cible.

    - Avec la stratégie d'approbation par défaut **Automatic**, OpenShift Virtualization se met automatiquement à jour vers la version correspondante après la mise à jour d'OpenShift Container Platform.
    - Si vous utilisez la stratégie d'approbation **Manual**, approuvez les mises à jour en attente en utilisant la console web.

13. Surveillez la mise à jour d'OpenShift Virtualization en exécutant la commande suivante :

```
$ oc get csv -n openshift-cnv
```

    La mise à jour est terminée lorsque le champ **VERSION** correspond à la version EUS cible et que le champ **PHASE** indique **Succeeded**.

14. Rétablissez la configuration des méthodes de mise à jour de la charge de travail que vous avez sauvegardée :

```
$ oc patch hco kubevirt-hyperconverged -n openshift-cnv --type json -p "
[{\"op\":\"add\",\"path\":\"/spec/workloadUpdateStrategy/workloadUpdateMethods\",
\"value\":$WORKLOAD_UPDATE_METHODS}]"
```

    Exemple de sortie

```
hyperconverged.hco.kubevirt.io/kubevirt-hyperconverged patched
```

    Vérification

    - Vérifiez l'état de la migration de la VM en exécutant la commande suivante :

```
$ oc get vmim -A
```

Prochaines étapes

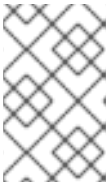
- Vous pouvez maintenant débloquer les pools de configuration des machines des nœuds de travail.

7.3. CONFIGURATION DES MÉTHODES DE MISE À JOUR DE LA CHARGE DE TRAVAIL

Vous pouvez configurer les méthodes de mise à jour de la charge de travail en modifiant la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- Pour utiliser la migration en direct comme méthode de mise à jour, vous devez d'abord activer la migration en direct dans le cluster.



NOTE

Si une CR **VirtualMachineInstance** contient **evictionStrategy: LiveMigrate** et que l'instance de machine virtuelle (VMI) ne prend pas en charge la migration en direct, la VMI ne sera pas mise à jour.

Procédure

1. Pour ouvrir le CR **HyperConverged** dans votre éditeur par défaut, exécutez la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Modifiez la strophe **workloadUpdateStrategy** du CR **HyperConverged**. Par exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  workloadUpdateStrategy:
    workloadUpdateMethods: 1
    - LiveMigrate 2
    - Evict 3
    batchEvictionSize: 10 4
    batchEvictionInterval: "1m0s" 5
...
```

1. Les méthodes qui peuvent être utilisées pour effectuer des mises à jour automatisées de la charge de travail. Les valeurs disponibles sont **LiveMigrate** et **Evict**. Si vous activez les deux options comme indiqué dans cet exemple, les mises à jour utilisent **LiveMigrate** pour les IMV qui prennent en charge la migration en direct et **Evict** pour toutes les IMV qui ne prennent pas en charge la migration en direct. Pour désactiver les mises à jour automatiques de la charge de travail, vous pouvez soit supprimer la strophe **workloadUpdateStrategy**, soit définir **workloadUpdateMethods: []** pour laisser le tableau vide.

2. Méthode de mise à jour la moins perturbatrice. Les IMV qui prennent en charge la migration en direct sont mises à jour en migrant l'invité de la machine virtuelle (VM) dans un nouveau pod avec les composants mis à jour activés. Si **LiveMigrate** est la seule méthode de mise à jour de la charge de travail répertoriée, les IMV qui ne prennent pas en charge la migration en direct ne sont pas perturbées ni mises à jour.

3. Méthode perturbatrice qui arrête les pods VMI pendant la mise à niveau. **Evict** est la seule méthode de mise à jour disponible si la migration en direct n'est pas activée dans le cluster. Si une IMV est contrôlée par un objet **VirtualMachine** qui a été configuré sur **runStrategy: always**, une nouvelle IMV est créée dans un nouveau pod avec des composants mis à jour.

4. Nombre d'IMV que l'on peut forcer à être mis à jour à la fois en utilisant la méthode **Evict**. Ceci ne s'applique pas à la méthode **LiveMigrate**.

5. L'intervalle à attendre avant d'expulser le prochain lot de charges de travail. Ceci ne s'applique pas à la méthode **LiveMigrate**.

NOTE

Vous pouvez configurer les limites et les délais de migration en direct en modifiant la strophe **spec.liveMigrationConfig** de la CR **HyperConverged**.

3. Pour appliquer vos modifications, enregistrez et quittez l'éditeur.

7.4. APPROBATION DES MISES À JOUR DE L'OPÉRATEUR EN ATTENTE

7.4.1. Approbation manuelle d'une mise à jour de l'opérateur en attente

Si la stratégie d'approbation de l'abonnement d'un opérateur installé est définie sur **Manual**, lorsque de nouvelles mises à jour sont publiées dans son canal de mise à jour actuel, la mise à jour doit être approuvée manuellement avant que l'installation ne puisse commencer.

Conditions préalables

- Un opérateur précédemment installé à l'aide de l'outil Operator Lifecycle Manager (OLM).

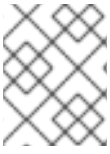
Procédure

1. Dans la perspective **Administrator** de la console web OpenShift Container Platform, naviguez vers **Operators → Installed Operators**.
2. Les opérateurs dont la mise à jour est en cours affichent un statut avec **Upgrade available**. Cliquez sur le nom de l'opérateur que vous souhaitez mettre à jour.
3. Cliquez sur l'onglet **Subscription**. Toute mise à jour nécessitant une approbation est affichée à côté de **Upgrade Status**. Par exemple, il peut s'agir de **1 requires approval**.
4. Cliquez sur **1 requires approval**, puis sur **Preview Install Plan**.
5. Examinez les ressources répertoriées comme étant disponibles pour une mise à jour. Lorsque vous êtes satisfait, cliquez sur **Approve**.
6. Retournez à la page **Operators → Installed Operators** pour suivre la progression de la mise à jour. Une fois la mise à jour terminée, le statut passe à **Succeeded** et **Up to date**.

7.5. SUIVI DE L'ÉTAT DE LA MISE À JOUR

7.5.1. Surveillance de l'état des mises à jour d'OpenShift Virtualization

Pour surveiller l'état d'une mise à niveau d'OpenShift Virtualization Operator, surveillez la version du service de cluster (CSV) **PHASE**. Vous pouvez également surveiller les conditions CSV dans la console web ou en exécutant la commande fournie ici.



NOTE

Les valeurs de **PHASE** et des conditions sont des approximations basées sur les informations disponibles.

Conditions préalables

- Connectez-vous au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Installez le CLI OpenShift (**oc**).

Procédure

1. Exécutez la commande suivante :

```
$ oc get csv -n openshift-cnv
```

2. Examinez le résultat, en vérifiant le champ **PHASE**. Par exemple :

Exemple de sortie

```
VERSION  REPLACES                                   PHASE
4.9.0    kubevirt-hyperconverged-operator.v4.8.2   Installing
4.9.0    kubevirt-hyperconverged-operator.v4.9.0   Replacing
```

3. Facultatif : Surveillez l'état agrégé de toutes les conditions du composant OpenShift Virtualization en exécutant la commande suivante :

```
$ oc get hco -n openshift-cnv kubevirt-hyperconverged \
-o=jsonpath='{range .status.conditions[*]}{.type}{"\t"}{.status}{"\t"}{.message}{"\n"}{end}'
```

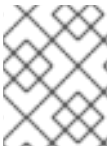
Une mise à niveau réussie donne le résultat suivant :

Exemple de sortie

```
ReconcileComplete  True  Reconcile completed successfully
Available          True  Reconcile completed successfully
Progressing        False Reconcile completed successfully
Degraded           False Reconcile completed successfully
Upgradeable        True  Reconcile completed successfully
```

7.5.2. Visualisation des charges de travail OpenShift Virtualization obsolètes

Vous pouvez afficher une liste des charges de travail obsolètes à l'aide de l'interface CLI.



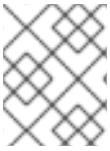
NOTE

Si votre cluster contient des pods de virtualisation obsolètes, l'alerte **OutdatedVirtualMachineInstanceWorkloads** se déclenche.

Procédure

- Pour afficher une liste des instances de machines virtuelles (VMI) obsolètes, exécutez la commande suivante :

```
$ oc get vmi -l kubevirt.io/outdatedLauncherImage --all-namespaces
```



NOTE

Configurer les mises à jour de la charge de travail pour s'assurer que les VMIs se mettent à jour automatiquement.

7.6. RESSOURCES SUPPLÉMENTAIRES

- Préparation d'une mise à jour EUS-to-EUS
- Qu'est-ce qu'un opérateur ?
- Concepts et ressources du gestionnaire du cycle de vie de l'opérateur
- Versions des services de cluster (CSV)
- Migration en direct de la machine virtuelle
- Configuration de la stratégie d'éviction des machines virtuelles
- Configuration des limites et des délais de migration en direct

CHAPITRE 8. POLITIQUES DE SÉCURITÉ

Les charges de travail des machines virtuelles (VM) s'exécutent en tant que pods non privilégiés. Pour que les VM puissent utiliser les fonctionnalités de virtualisation d'OpenShift, certains pods bénéficient de politiques de sécurité personnalisées qui ne sont pas disponibles pour les autres propriétaires de pods :

- Une politique SELinux étendue **container_t** s'applique aux pods **virt-launcher**.
- Des contraintes de contexte de sécurité (SCC) sont définies pour le compte de service **kubevirt-controller**.

8.1. SÉCURITÉ DE LA CHARGE DE TRAVAIL

Par défaut, les charges de travail des machines virtuelles (VM) ne s'exécutent pas avec les privilèges root dans OpenShift Virtualization.

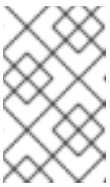
Pour chaque VM, un pod **virt-launcher** exécute une instance de **libvirt** dans *session mode* pour gérer le processus de la VM. En mode session, le démon **libvirt** s'exécute en tant que compte d'utilisateur non root et n'autorise que les connexions des clients qui s'exécutent sous le même identifiant d'utilisateur (UID). Par conséquent, les VM s'exécutent en tant que pods non privilégiés, conformément au principe de sécurité du moindre privilège.

Il n'y a pas de fonctionnalités OpenShift Virtualization supportées qui requièrent des privilèges root. Si une fonctionnalité nécessite des privilèges root, il se peut qu'elle ne soit pas supportée pour une utilisation avec OpenShift Virtualization.

8.2. POLITIQUES SELINUX ÉTENDUES POUR LES PODS VIRT-LAUNCHER

La politique SELinux de **container_t** pour les pods **virt-launcher** est étendue pour permettre les fonctions essentielles d'OpenShift Virtualization.

- La stratégie suivante est requise pour la mise en place d'une file d'attente réseau, ce qui permet d'adapter les performances du réseau à l'augmentation du nombre de vCPU disponibles :
  - **allow process self (tun_socket (relabelfrom relabelto attach_queue))**
- La politique suivante autorise **virt-launcher** à lire les fichiers du répertoire **/proc**, y compris **/proc/cpuinfo** et **/proc/uptime**:
  - **allow process proc_type (file (getattr open read))**
- La stratégie suivante autorise **libvirtd** à relayer les messages de débogage liés au réseau.
  - **allow process self (netlink_audit_socket (nlmsg_relay))**



NOTE

Sans cette politique, toute tentative de relayer les messages de débogage du réseau est bloquée. Cela pourrait remplir les journaux d'audit du nœud avec des refus SELinux.

- Les règles suivantes permettent à **libvirtd** d'accéder à **hugetblfs**, ce qui est nécessaire pour prendre en charge les pages volumineuses :

- **allow process hugetlbfs_t (dir (add_name create write remove_name rmdir setattr))**
- **allow process hugetlbfs_t (file (create unlink))**

- Les stratégies suivantes permettent à **virtiofs** de monter des systèmes de fichiers et d'accéder à NFS :
  - **allow process nfs_t (dir (mounton))**
  - **allow process proc_t (dir (mounton))**
  - **allow process proc_t (filesystem (mount unmount))**
- La politique suivante est héritée de Kubevirt en amont, où elle active le réseau **passt**:
  - **allow process tmpfs_t (filesystem (mount))**

NOTE

OpenShift Virtualization ne prend pas en charge **passt** pour le moment.

8.3. CONTRAINTES DE CONTEXTE DE SÉCURITÉ ET CAPACITÉS LINUX SUPPLÉMENTAIRES DE OPENSHIFT CONTAINER PLATFORM POUR LE COMPTE DE SERVICE KUBEVIRT-CONTROLLER

Les contraintes de contexte de sécurité (SCC) contrôlent les autorisations pour les modules. Ces autorisations comprennent les actions qu'un module, un ensemble de conteneurs, peut effectuer et les ressources auxquelles il peut accéder. Vous pouvez utiliser les contraintes de contexte de sécurité pour définir un ensemble de conditions qu'un module doit respecter pour être accepté dans le système.

virt-controller est un contrôleur de cluster qui crée les pods **virt-launcher** pour les machines virtuelles dans le cluster. Ces pods sont autorisés par le compte de service **kubevirt-controller**.

Le compte de service **kubevirt-controller** se voit attribuer des SCC et des capacités Linux supplémentaires afin de pouvoir créer des pods **virt-launcher** avec les autorisations appropriées. Ces autorisations étendues permettent aux machines virtuelles d'utiliser les fonctionnalités d'OpenShift Virtualization qui dépassent la portée des pods typiques.

Le compte de service **kubevirt-controller** dispose des SCC suivants :

- **scc.AllowHostDirVolumePlugin = true**
  Cela permet aux machines virtuelles d'utiliser le plugin de volume hostpath.
- **scc.AllowPrivilegedContainer = false**
  Cela permet de s'assurer que le pod virt-launcher n'est pas exécuté en tant que conteneur privilégié.
- **scc.AllowedCapabilities = []corev1.Capability{"SYS_NICE", "NET_BIND_SERVICE", "SYS_PTRACE"}**
  - **SYS_NICE** permet de définir l'affinité de l'unité centrale.
  - **NET_BIND_SERVICE** permet les opérations DHCP et Slirp.
  - **SYS_PTRACE** permet à certaines versions de **libvirt** de trouver l'identifiant de processus (PID) de **swtpm**, un émulateur logiciel de Trusted Platform Module (TPM).

8.3.1. Affichage des définitions SCC et RBAC pour le contrôleur kubevirt

Vous pouvez consulter la définition de **SecurityContextConstraints** pour **kubevirt-controller** en utilisant l'outil **oc**:

```
$ oc get scc kubevirt-controller -o yaml
```

Vous pouvez voir la définition RBAC pour le rôle de cluster **kubevirt-controller** en utilisant l'outil **oc**:

```
$ oc get clusterrole kubevirt-controller -o yaml
```

8.4. RESSOURCES SUPPLÉMENTAIRES

- Gestion des contraintes liées au contexte de sécurité
- Utilisation de RBAC pour définir et appliquer des autorisations
- Optimisation des performances réseau des machines virtuelles dans la documentation de Red Hat Enterprise Linux (RHEL)
- Utilisation de pages volumineuses avec des machines virtuelles
- Configuration de pages volumineuses dans la documentation RHEL

CHAPITRE 9. UTILISATION DES OUTILS CLI

Les deux principaux outils CLI utilisés pour gérer les ressources dans le cluster sont les suivants :

- Le client OpenShift Virtualization **virtctl**
- Le client OpenShift Container Platform **oc**

9.1. CONDITIONS PRÉALABLES

- Vous devez installer le client **virtctl** .

9.2. COMMANDES DU CLIENT OPENSHIFT CONTAINER PLATFORM

Le client **oc** d'OpenShift Container Platform est un utilitaire de ligne de commande permettant de gérer les ressources d'OpenShift Container Platform, notamment les types d'objets **VirtualMachine** (**vm**) et **VirtualMachineInstance** (**vmi**).

NOTE

Vous pouvez utiliser l'option **-n <namespace>** pour spécifier un projet différent.

Tableau 9.1. **oc** commandes

Commandement	Description
oc login -u <user_name>	Connectez-vous au cluster OpenShift Container Platform en tant que **<user_name>**.
oc get <object_type>	Affiche une liste d'objets pour le type d'objet spécifié dans le projet en cours.
oc describe <object_type> <resource_name>	Affiche les détails de la ressource spécifique dans le projet en cours.
oc create -f <object_config>	Créer une ressource dans le projet en cours à partir d'un nom de fichier ou de stdin.
oc edit <object_type> <resource_name>	Modifier une ressource dans le projet en cours.
oc delete <object_type> <resource_name>	Supprimer une ressource dans le projet en cours.

Pour plus d'informations sur les commandes du client **oc**, voir la documentation sur les outils CLI de OpenShift Container Platform.

9.3. COMMANDES VIRTCTL

Le client **virtctl** est un utilitaire de ligne de commande permettant de gérer les ressources de virtualisation OpenShift.

Tableau 9.2. virtctl commandements généraux

Commandement	Description
virtctl version	Consultez les versions du client et du serveur **virtctl**.
virtctl help	Voir la liste des commandes **virtctl**.
virtctl <command> -h|--help	Affiche une liste d'options pour une commande spécifique.
virtctl options	Affiche la liste des options de commande globales pour n'importe quelle commande **virtctl**.

9.3.1. Commandes de gestion des VM et VMI

Vous pouvez utiliser **virtctl** pour gérer l'état des machines virtuelles (VM) ou des instances de machines virtuelles (VMI) et pour migrer une VM.

Tableau 9.3. virtctl Commandes de gestion des machines virtuelles

Commandement	Description
virtctl start <vm_name>	Démarrer une VM.
virtctl start --paused <vm_name>	Démarrer une VM en état de pause. Cette option permet d'interrompre le processus de démarrage à partir de la console VNC.
virtctl stop <vm_name>	Arrêter une VM.
virtctl stop <vm_name> --grace-period 0 --force	Forcer l'arrêt d'une VM. Cette option peut entraîner des incohérences ou des pertes de données.
virtctl pause vm|vmi <vm_name>	Mettre en pause une VM ou une VMI. L'état de la machine est conservé en mémoire.
virtctl unpause vm|vmi <vm_name>	Désactiver la pause d'une VM ou d'une VMI.
virtctl migrate <vm_name>	Migrer une VM.
virtctl restart <vm_name>	Redémarrer une VM.

9.3.2. Commandes de connexion VM et VMI

Vous pouvez utiliser **virtctl** pour vous connecter à la console série, exposer un port, établir une connexion proxy, spécifier un port et ouvrir une connexion VNC à une VM.

Tableau 9.4. **virtctl console**les commandes**expose** et **vnc**

Commandement	Description
virtctl console <vmi_name>	Se connecter à la console série d'un VMI.
virtctl expose <vm_name>	Créer un service qui transfère un port désigné d'une VM ou d'une VMI et exposer le service sur le port spécifié du nœud.
virtctl vnc --kubeconfig=$KUBECONFIG <vmi_name>	Ouvrez une connexion VNC (Virtual Network Client) à une IMV. L'accès à la console graphique d'une IMV via VNC nécessite un visualiseur à distance sur votre machine locale.
virtctl vnc --kubeconfig=$KUBECONFIG --proxy-only=true <vmi_name>	Affichez le numéro de port et connectez-vous manuellement à une IMV en utilisant n'importe quelle visionneuse via la connexion VNC.
virtctl vnc --kubeconfig=$KUBECONFIG --port=<port-number> <vmi_name>	Spécifiez un numéro de port pour exécuter le proxy sur le port spécifié, si ce port est disponible. Si aucun numéro de port n'est spécifié, le proxy fonctionne sur un port aléatoire.

9.3.3. Commandes d'exportation de volumes VM

Vous pouvez utiliser les commandes **virtctl vmexport** pour créer, télécharger ou supprimer un volume exporté à partir d'une VM, d'un snapshot VM ou d'une revendication de volume persistant (PVC).

Tableau 9.5. **virtctl vmexport** commandes

Commandement	Description
virtctl vmexport create <vmexport_name> --vm|snapshot|pvc= <object_name>	Créez une ressource personnalisée (CR) **VirtualMachineExport** pour exporter un volume à partir d'une VM, d'un snapshot VM ou d'un PVC. • **--vm**: Exporte les PVC d'une VM. • **--snapshot**: Exporte les PVC contenus dans un CR **VirtualMachineSnapshot**. • **--pvc**: Exporte un PVC. • Facultatif : **--ttl=1h** indique la durée de vie. La durée par défaut est de 2 heures.
virtctl vmexport delete <vmexport_name>	Supprimer manuellement un CR **VirtualMachineExport**.

Commandement	Description
virtctl vmexport download <vmexport_name> --output= <output_file> --volume= <volume_name>	Télécharger le volume défini dans un CR **VirtualMachineExport**. • **--output** spécifie le format du fichier. Exemple :**disk.img.gz**. • **--volume** spécifie le volume à télécharger. Cette option est facultative si un seul volume est disponible. En option : • **--keep-vme** conserve le CR **VirtualMachineExport** après le téléchargement. Le comportement par défaut est de supprimer le CR **VirtualMachineExport** après le téléchargement. • **--insecure** active une connexion HTTP non sécurisée.
virtctl vmexport download <vmexport_name> -- <vm|snapshot|pvc>= <object_name> --output= <output_file> --volume= <volume_name>	Créez un CR **VirtualMachineExport** et téléchargez ensuite le volume défini dans le CR.

9.3.4. Commandes de vidage de la mémoire de la VM

Vous pouvez utiliser la commande **virtctl memory-dump** pour obtenir un vidage de la mémoire d'une machine virtuelle (VM) sur un PVC. Vous pouvez spécifier un PVC existant ou utiliser l'option **--create-claim** pour créer un nouveau PVC.

Conditions préalables

- Le mode de volume du PVC doit être **FileSystem**.
- Le PVC doit être suffisamment grand pour contenir le vidage de la mémoire.
  The formula for calculating the PVC size is **(VMMemorySize + 100Mi) * FileSystemOverhead**, where **100Mi** is the memory dump overhead.
- Vous devez activer la porte de la fonctionnalité hot plug dans la ressource personnalisée **HyperConverged** en exécutant la commande suivante :

```
$ oc patch hco kubevirt-hyperconverged -n openshift-cnv \
  --type json -p '[{"op": "add", "path": "/spec/featureGates", \
  "value": "HotplugVolumes"}]'
```

Téléchargement du fichier mémoire

Vous devez utiliser la commande **virtctl vmexport download** pour télécharger le fichier mémoire :

```
$ virtctl vmexport download <vmexport_name> --vm\|pvc=<object_name> \
  --volume=<volume_name> --output=<output_file>
```

Tableau 9.6. **virtctl memory-dump** commandes

Commandement	Description
virtctl memory-dump get <vm_name> --claim-name= <pvc_name>	Sauvegarder le vidage de la mémoire d'une VM sur un PVC. L'état du vidage de la mémoire est affiché dans la section **status** de la ressource **VirtualMachine**. En option : ● **--create-claim** crée un nouveau PVC de la taille appropriée. Cet indicateur a les options suivantes : ○ **--storage-class=<storage_class>**: Spécifier une classe de stockage pour le PVC. ○ **--access-mode=<access_mode>**: Indiquez **ReadWriteOnce** ou **ReadWriteMany**.
virtctl memory-dump get <vm_name>	Réexécutez la commande **virtctl memory-dump** avec le même PVC. Cette commande écrase la précédente vidange de la mémoire.
virtctl memory-dump remove <vm_name>	Supprimer un vidage de la mémoire. Vous devez supprimer manuellement un vidage de mémoire si vous souhaitez modifier le PVC cible. Cette commande supprime l'association entre la VM et le PVC, de sorte que le vidage de la mémoire n'est pas affiché dans la section **status** de la ressource **VirtualMachine**. Le PVC n'est pas affecté.

9.3.5. Commandes de téléchargement d'images

Vous pouvez utiliser les commandes **virtctl image-upload** pour télécharger une image VM sur un volume de données.

Tableau 9.7. **virtctl image-upload** commandes

Commandement	Description
virtctl image-upload dv <datavolume_name> --image-path= </path/to/image> --no-create	Télécharger une image de VM sur un volume de données existant.
virtctl image-upload dv <datavolume_name> --size= <datavolume_size> --image-path=</path/to/image>	Téléchargement d'une image de VM dans un nouveau volume de données d'une taille spécifiée.

9.3.6. Commandes d'informations sur l'environnement

Vous pouvez utiliser **virtctl** pour afficher des informations sur les versions, les systèmes de fichiers, les systèmes d'exploitation invités et les utilisateurs connectés.

Tableau 9.8. virtctl commandes d'information sur l'environnement

Commandement	Description
virtctl fslist <vmi_name>	Afficher les systèmes de fichiers disponibles sur une machine invitée.
virtctl guestosinfo <vmi_name>	Afficher des informations sur les systèmes d'exploitation d'une machine invitée.
virtctl userlist <vmi_name>	Afficher les utilisateurs connectés sur une machine invitée.

9.4. CRÉATION D'UN CONTENEUR À L'AIDE DE VIRTCTL GUESTFS

Vous pouvez utiliser la commande **virtctl guestfs** pour déployer un conteneur interactif avec **libguestfs-tools** et une revendication de volume persistant (PVC) qui lui est attachée.

Procédure

- Pour déployer un conteneur avec **libguestfs-tools**, monter le PVC et y attacher un shell, exécutez la commande suivante :

```
virtctl guestfs -n <namespace> <pvc_name> 1
```

1. Le nom du PVC est un argument obligatoire. Si vous ne l'incluez pas, un message d'erreur apparaît.

9.5. OUTILS LIBGUESTFS ET VIRTCTL GUESTFS

Libguestfs vous aident à accéder aux images de disques de machines virtuelles (VM) et à les modifier. Vous pouvez utiliser les outils **libguestfs** pour afficher et modifier des fichiers dans un invité, cloner et construire des machines virtuelles, et formater et redimensionner des disques.

Vous pouvez également utiliser la commande **virtctl guestfs** et ses sous-commandes pour modifier, inspecter et déboguer les disques VM sur un PVC. Pour obtenir une liste complète des sous-commandes possibles, entrez **virt-** dans la ligne de commande et appuyez sur la touche Tab. Par exemple :

Commandement	Description
virt-edit -a /dev/vda /etc/motd	Modifier un fichier de manière interactive dans votre terminal.
virt-customize -a /dev/vda --ssh-inject root:string:<public key example>	Injecter une clé ssh dans l'invité et créer un login.
virt-df -a /dev/vda -h	Voir combien d'espace disque est utilisé par une VM.

Commandement	Description
virt-customize -a /dev/vda --run-command 'rpm -qa > /rpm-list'	Voir la liste complète de tous les RPM installés sur un invité en créant un fichier de sortie contenant la liste complète.
virt-cat -a /dev/vda /rpm-list	Affiche la liste des fichiers de sortie de tous les RPM créés à l'aide de la commande **virt-customize -a /dev/vda --run-command 'rpm -qa > /rpm-list'** dans votre terminal.
virt-sysprep -a /dev/vda	Scelle une image de disque de machine virtuelle à utiliser comme modèle.

Par défaut, **virtctl guestfs** crée une session avec tout ce qui est nécessaire pour gérer le disque d'une VM. Toutefois, la commande prend également en charge plusieurs options de drapeaux si vous souhaitez personnaliser le comportement :

Option drapeau	Description
--h ou **--help**	Fournit de l'aide pour **guestfs**.
-n <namespace> avec un argument **<pvc_name>**	Pour utiliser un PVC d'un espace de noms spécifique. Si vous n'utilisez pas l'option **-n <namespace>**, votre projet actuel est utilisé. Pour changer de projet, utilisez l'option **oc project <namespace>**. Si vous n'incluez pas l'argument **<pvc_name>**, un message d'erreur apparaît.
--image string	Liste l'image du conteneur **libguestfs-tools**. Vous pouvez configurer le conteneur pour qu'il utilise une image personnalisée en utilisant l'option **--image**.
--kvm	Indique que **kvm** est utilisé par le conteneur **libguestfs-tools**. Par défaut, **virtctl guestfs** met en place **kvm** pour le conteneur interactif, ce qui accélère considérablement l'exécution de **libguest-tools** car il utilise QEMU. Si un cluster n'a pas de nœuds de support **kvm**, vous devez désactiver **kvm** en définissant l'option **--kvm=false**. S'il n'est pas défini, le pod **libguestfs-tools** reste en attente car il ne peut être planifié sur aucun nœud.
--pull-policy string	Affiche la politique d'extraction de l'image **libguestfs**. Vous pouvez également écraser la politique d'extraction de l'image en définissant l'option **pull-policy**.

La commande vérifie également si un PVC est utilisé par un autre pod, auquel cas un message d'erreur apparaît. Cependant, une fois que le processus **libguestfs-tools** a démarré, l'installation ne peut pas éviter qu'un nouveau pod utilise le même PVC. Vous devez vérifier qu'il n'y a pas de pods **virtctl guestfs** actifs avant de démarrer la VM qui accède au même PVC.

NOTE

La commande **virtctl guestfs** n'accepte qu'un seul PVC attaché au module interactif.

9.6. RESSOURCES SUPPLÉMENTAIRES

- Libguestfs : outils permettant d'accéder aux images de disques de machines virtuelles et de les modifier.

CHAPITRE 10. MACHINES VIRTUELLES

10.1. CRÉATION DE MACHINES VIRTUELLES

Utilisez l'une des procédures suivantes pour créer une machine virtuelle :

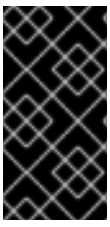
- Visite guidée de démarrage rapide
- Création rapide à partir du **Catalog**
- Coller un fichier YAML préconfiguré avec l'assistant de machine virtuelle
- Utilisation de l'interface de programmation

AVERTISSEMENT

Ne créez pas de machines virtuelles dans les espaces de noms **openshift-***. Créez plutôt un nouvel espace de noms ou utilisez un espace de noms existant sans le préfixe **openshift**.

Lorsque vous créez des machines virtuelles à partir de la console Web, sélectionnez un modèle de machine virtuelle configuré avec une source de démarrage. Les modèles de machine virtuelle avec une source de démarrage sont étiquetés comme **Available boot source** ou ils affichent un texte d'étiquette personnalisé. L'utilisation de modèles avec une source de démarrage disponible accélère le processus de création de machines virtuelles.

Les modèles sans source de démarrage sont étiquetés comme **Boot source required**. Vous pouvez utiliser ces modèles si vous suivez les étapes pour ajouter une source de démarrage à la machine virtuelle.



IMPORTANT

En raison de différences dans le comportement de stockage, certains modèles de machines virtuelles sont incompatibles avec OpenShift à nœud unique. Pour assurer la compatibilité, ne définissez pas le champ **evictionStrategy** pour les modèles ou les machines virtuelles qui utilisent des volumes de données ou des profils de stockage.

10.1.1. Utilisation d'un démarrage rapide pour créer une machine virtuelle

La console Web fournit des démarrages rapides avec des visites guidées pour la création de machines virtuelles. Vous pouvez accéder au catalogue Quick Starts en sélectionnant le menu Help dans la perspective **Administrator** pour afficher le catalogue Quick Starts. Lorsque vous cliquez sur une tuile de démarrage rapide et que vous commencez la visite, le système vous guide tout au long du processus.

Les tâches d'un démarrage rapide commencent par la sélection d'un modèle Red Hat. Ensuite, vous pouvez ajouter une source de démarrage et importer l'image du système d'exploitation. Enfin, vous pouvez enregistrer le modèle personnalisé et l'utiliser pour créer une machine virtuelle.

Conditions préalables

- Accès au site web où vous pouvez télécharger le lien URL de l'image du système d'exploitation.

Procédure

1. Dans la console web, sélectionnez **Quick Starts** dans le menu Aide.
2. Cliquez sur une tuile du catalogue Quick Starts. Par exemple : **Creating a Red Hat Linux Enterprise Linux virtual machine**.
3. Suivez les instructions de la visite guidée et effectuez les tâches d'importation d'une image de système d'exploitation et de création d'une machine virtuelle. La page **Virtualization → VirtualMachines** affiche la machine virtuelle.

10.1.2. Création rapide d'une machine virtuelle

Vous pouvez créer rapidement une machine virtuelle (VM) en utilisant un modèle avec une source de démarrage disponible.

Procédure

1. Cliquez sur **Virtualization → Catalog** dans le menu latéral.
2. Cliquez sur **Boot source available** pour filtrer les modèles avec les sources de démarrage.

NOTE

Par défaut, la liste des modèles n'affiche que **Default Templates**. Cliquez sur **All Items** lors du filtrage pour afficher tous les modèles disponibles pour les filtres que vous avez choisis.

3. Cliquez sur un modèle pour en afficher les détails.
4. Cliquez sur **Quick Create VirtualMachine** pour créer une VM à partir du modèle.
   La page de la machine virtuelle **Details** s'affiche avec l'état du provisionnement.

Vérification

1. Cliquez sur **Events** pour afficher un flux d'événements au fur et à mesure que la VM est approvisionnée.
2. Cliquez sur **Console** pour vérifier que la VM a bien démarré.

10.1.3. Création d'une machine virtuelle à partir d'un modèle personnalisé

Certains modèles nécessitent des paramètres supplémentaires, par exemple un PVC avec une source de démarrage. Vous pouvez personnaliser certains paramètres d'un modèle pour créer une machine virtuelle (VM).

Procédure

1. Dans la console web, sélectionnez un modèle :
   a. Cliquez sur **Virtualization → Catalog** dans le menu latéral.

b. Facultatif : Filtrez les modèles par projet, mot-clé, système d'exploitation ou profil de charge de travail.

c. Cliquez sur le modèle que vous souhaitez personnaliser.

2. Cliquez sur **Customize VirtualMachine**.

3. Spécifiez les paramètres de votre VM, y compris ses adresses **Name** et **Disk source**. Vous pouvez éventuellement spécifier une source de données à cloner.

Vérification

1. Cliquez sur **Events** pour afficher un flux d'événements au fur et à mesure que la VM est approvisionnée.

2. Cliquez sur **Console** pour vérifier que la VM a bien démarré.

Reportez-vous à la section relative aux champs de la machine virtuelle lors de la création d'une VM à partir de la console web.

10.1.3.1. Domaines de mise en réseau

Nom	Description
Nom	Nom du contrôleur d'interface réseau.
Model	Indique le modèle du contrôleur d'interface réseau. Les valeurs prises en charge sont **e1000e** et **virtio**.
Réseau	Liste des définitions de pièces jointes disponibles.
Type	Liste des méthodes de liaison disponibles. Sélectionnez la méthode de liaison adaptée à l'interface réseau : ● Réseau de pods par défaut : **masquerade** ● Réseau de ponts Linux : **bridge** ● Réseau SR-IOV : **SR-IOV**
Adresse MAC	Adresse MAC du contrôleur d'interface réseau. Si aucune adresse MAC n'est spécifiée, une adresse est attribuée automatiquement.

10.1.3.2. Champs de stockage

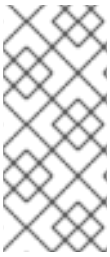
Nom	La sélection	Description
Source	Vierge (crée du PVC)	Créer un disque vide.

Nom	La sélection	Description
	Importation par URL (crée un PVC)	Importer du contenu via une URL (HTTP ou HTTPS).
	Utiliser un PVC existant	Utiliser un PVC déjà disponible dans le cluster.
	Cloner un PVC existant (crée un PVC)	Sélectionnez un PVC existant disponible dans le cluster et clonez-le.
	Importation via le registre (création de PVC)	Importer du contenu via le registre des conteneurs.
	Conteneur (éphémère)	Télécharger le contenu d'un conteneur situé dans un registre accessible depuis le cluster. Le disque conteneur ne doit être utilisé que pour les systèmes de fichiers en lecture seule, tels que les CD-ROM ou les machines virtuelles temporaires.
Nom		Nom du disque. Le nom peut contenir des lettres minuscules (**a-z**), des chiffres (**0-9**), des traits d'union (**-**) et des points (**.**), jusqu'à un maximum de 253 caractères. Le premier et le dernier caractères doivent être alphanumériques. Le nom ne doit pas contenir de majuscules, d'espaces ou de caractères spéciaux.
Taille		Taille du disque en gigaoctets.
Type		Type de disque. Exemple : Disque ou CD-ROM
Interface		Type de périphérique de disque. Les interfaces prises en charge sont **virtIO**, **SATA** et **SCSI**.
Classe de stockage		La classe de stockage utilisée pour créer le disque.

Paramètres de stockage avancés

Les paramètres de stockage avancés suivants sont facultatifs et disponibles pour les disques **Blank**, **Import via URL**, et **Clone existing PVC**. Avant OpenShift Virtualization 4.11, si vous ne spécifiez pas ces paramètres, le système utilise les valeurs par défaut de la carte de configuration **kubevirt-storage-**

class-defaults. Dans OpenShift Virtualization 4.11 et les versions ultérieures, le système utilise les valeurs par défaut du profil de stockage .



NOTE

Utilisez les profils de stockage pour garantir des paramètres de stockage avancés cohérents lors du provisionnement du stockage pour OpenShift Virtualization.

Pour spécifier manuellement **Volume Mode** et **Access Mode**, vous devez décocher la case **Apply optimized StorageProfile settings**, qui est sélectionnée par défaut.

Nom	Description du mode	Paramètres	Description des paramètres
Mode volume	Définit si le volume persistant utilise un système de fichiers formaté ou un état de bloc brut. La valeur par défaut est **Filesystem**.	Système de fichiers	Stocke le disque virtuel sur un volume basé sur un système de fichiers.
		Bloc	Enregistre le disque virtuel directement sur le volume de blocs. N'utilisez **Block** que si le stockage sous-jacent le prend en charge.
Mode d'accès	Mode d'accès au volume persistant.	ReadWriteOnce (RWO)	Le volume peut être monté en lecture-écriture par un seul nœud.
		ReadWriteMany (RWX)	Le volume peut être monté en lecture-écriture par plusieurs nœuds à la fois.  NOTE Cela est nécessaire pour certaines fonctionnalités , telles que la migration en direct des machines virtuelles entre les nœuds.
		ReadOnlyMany (ROX)	Le volume peut être monté en lecture seule par plusieurs nœuds.

10.1.3.3. Champs d'initialisation du nuage

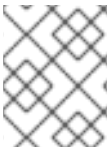
Nom	Description
Clés SSH autorisées	La clé publique de l'utilisateur qui est copiée sur ***~/.ssh/authorized_keys*** sur la machine virtuelle.
Script personnalisé	Remplace les autres options par un champ dans lequel vous pouvez coller un script personnalisé de démarrage du nuage.

Pour configurer les valeurs par défaut des classes de stockage, utilisez les profils de stockage. Pour plus d'informations, voir Personnaliser le profil de stockage .

10.1.3.4. Coller un fichier YAML préconfiguré pour créer une machine virtuelle

Créez une machine virtuelle en écrivant ou en collant un fichier de configuration YAML. Une configuration de machine virtuelle **example** valide est fournie par défaut chaque fois que vous ouvrez l'écran d'édition YAML.

Si votre configuration YAML n'est pas valide lorsque vous cliquez sur **Create**, un message d'erreur indique le paramètre dans lequel l'erreur se produit. Une seule erreur est affichée à la fois.



NOTE

Le fait de quitter l'écran YAML en cours d'édition annule toutes les modifications apportées à la configuration.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Cliquez sur **Create** et sélectionnez **With YAML**.
3. Écrivez ou collez votre configuration de machine virtuelle dans la fenêtre éditable.
    a. Vous pouvez également utiliser la machine virtuelle **example** fournie par défaut dans l'écran YAML.
4. Facultatif : Cliquez sur **Download** pour télécharger le fichier de configuration YAML dans son état actuel.
5. Cliquez sur **Create** pour créer la machine virtuelle.

La machine virtuelle est répertoriée sur la page **VirtualMachines**.

10.1.4. Utilisation de l'interface de programmation pour créer une machine virtuelle

Vous pouvez créer une machine virtuelle à partir d'un manifeste **virtualMachine**.

Procédure

1. Modifiez le manifeste **VirtualMachine** pour votre VM. Par exemple, le manifeste suivant configure une VM Red Hat Enterprise Linux (RHEL) :

    Exemple 10.1. Exemple de manifeste pour une VM RHEL

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    app: <vm_name> 1
  name: <vm_name>
spec:
  dataVolumeTemplates:
  - apiVersion: cdi.kubevirt.io/v1beta1
    kind: DataVolume
    metadata:
      name: <vm_name>
    spec:
      sourceRef:
        kind: DataSource
        name: rhel9
        namespace: openshift-virtualization-os-images
      storage:
        resources:
          requests:
            storage: 30Gi
  running: false
  template:
    metadata:
      labels:
        kubevirt.io/domain: <vm_name>
    spec:
      domain:
        cpu:
          cores: 1
          sockets: 2
          threads: 1
        devices:
          disks:
          - disk:
              bus: virtio
            name: rootdisk
          - disk:
              bus: virtio
            name: cloudinitdisk
          interfaces:
          - masquerade: {}
            name: default
          rng: {}
        features:
          smm:
            enabled: true
        firmware:
          bootloader:
            efi: {}
        resources:
          requests:
            memory: 8Gi
      evictionStrategy: LiveMigrate
      networks:
      - name: default
```

```
    pod: {}
  volumes:
  - dataVolume:
      name: <vm_name>
    name: rootdisk
  - cloudInitNoCloud:
      userData: |-
        #cloud-config
        user: cloud-user
        password: '<password>' 2
        chpasswd: { expire: False }
    name: cloudinitdisk
```

1 Indiquez le nom de la machine virtuelle.

2 Spécifiez le mot de passe pour cloud-user.

2. Créez une machine virtuelle en utilisant le fichier manifeste :

```
oc create -f <vm_manifest_file>.yaml
```

3. Facultatif : Démarrer la machine virtuelle :

```
virtctl start <vm_name>
```

10.1.5. Types de volumes de stockage de la machine virtuelle

Type de volume de stockage	Description
éphémère	Une image COW (copy-on-write) locale qui utilise un volume réseau comme magasin de sauvegarde en lecture seule. Le volume de sauvegarde doit être un **PersistentVolumeClaim**. L'image éphémère est créée au démarrage de la machine virtuelle et stocke toutes les écritures localement. L'image éphémère est supprimée lorsque la machine virtuelle est arrêtée, redémarrée ou supprimée. Le volume de sauvegarde (PVC) n'est en aucun cas modifié.
réclamation persistante sur les volumes	Attache un PV disponible à une machine virtuelle. L'attachement d'un PV permet aux données de la machine virtuelle de persister entre les sessions. L'importation d'un disque de machine virtuelle existant dans un PVC en utilisant CDI et en attachant le PVC à une instance de machine virtuelle est la méthode recommandée pour importer des machines virtuelles existantes dans OpenShift Container Platform. Certaines conditions sont requises pour que le disque puisse être utilisé dans un PVC.

Type de volume de stockage	Description
donnéesVolume	Les volumes de données s'appuient sur le type de disque **persistentVolumeClaim** en gérant le processus de préparation du disque de la machine virtuelle via une opération d'importation, de clonage ou de téléchargement. Les machines virtuelles qui utilisent ce type de volume sont assurées de ne pas démarrer tant que le volume n'est pas prêt. Spécifiez **type: dataVolume** ou **type: ""**. Si vous spécifiez une autre valeur pour **type**, telle que **persistentVolumeClaim**, un avertissement s'affiche et la machine virtuelle ne démarre pas.
cloudInitNoCloud	Attache un disque qui contient la source de données NoCloud référencée cloud-init, fournissant des données utilisateur et des métadonnées à la machine virtuelle. Une installation cloud-init est requise à l'intérieur du disque de la machine virtuelle.
containerDisk	Fait référence à une image, telle qu'un disque de machine virtuelle, qui est stockée dans le registre d'images du conteneur. L'image est extraite du registre et attachée à la machine virtuelle en tant que disque lorsque la machine virtuelle est lancée. Un volume **containerDisk** n'est pas limité à une seule machine virtuelle et est utile pour créer un grand nombre de clones de machines virtuelles qui ne nécessitent pas de stockage permanent. Seuls les formats RAW et QCOW2 sont des types de disques pris en charge pour le registre des images de conteneurs. Le format QCOW2 est recommandé pour réduire la taille de l'image.  **NOTE** Un volume **containerDisk** est éphémère. Il est supprimé lorsque la machine virtuelle est arrêtée, redémarrée ou supprimée. Un volume **containerDisk** est utile pour les systèmes de fichiers en lecture seule tels que les CD-ROM ou pour les machines virtuelles jetables.

Type de volume de stockage	Description
emptyDisk	Crée un disque QCOW2 clair supplémentaire lié au cycle de vie de l'interface de la machine virtuelle. Les données survivent aux redémarrages de la machine virtuelle initiés par l'invité, mais sont supprimées lorsque la machine virtuelle s'arrête ou est redémarrée à partir de la console web. Le disque vide est utilisé pour stocker les dépendances des applications et les données qui dépassent autrement le système de fichiers temporaires limité d'un disque éphémère. La taille du disque **capacity** doit également être indiquée.

10.1.6. À propos des stratégies d'exécution pour les machines virtuelles

Le paramètre **RunStrategy** pour les machines virtuelles détermine le comportement d'une instance de machine virtuelle (VMI) en fonction d'une série de conditions. Le paramètre **spec.runStrategy** existe dans le processus de configuration de la machine virtuelle en tant qu'alternative au paramètre **spec.running**. Le paramètre **spec.runStrategy** permet une plus grande flexibilité dans la création et la gestion des IMV, contrairement au paramètre **spec.running** qui ne permet que des réponses **true** ou **false**. Toutefois, les deux paramètres s'excluent mutuellement. Seuls **spec.running** ou **spec.runStrategy** peuvent être utilisés. Une erreur se produit si les deux sont utilisés.

Il existe quatre stratégies d'exécution définies.

Always

Une VMI est toujours présente lorsqu'une machine virtuelle est créée. Une nouvelle VMI est créée si l'originale s'arrête pour quelque raison que ce soit, ce qui correspond au comportement de **spec.running: true**.

RerunOnFailure

Une IMV est recréée si l'instance précédente échoue en raison d'une erreur. L'instance n'est pas recréée si la machine virtuelle s'arrête avec succès, par exemple lorsqu'elle s'éteint.

Manual

Les commandes client virtctl **start**, **stop**, et **restart** peuvent être utilisées pour contrôler l'état et l'existence de la VMI.

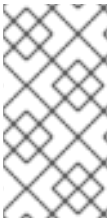
Halted

Aucune VMI n'est présente lors de la création d'une machine virtuelle, ce qui correspond au comportement de **spec.running: false**.

Différentes combinaisons des commandes virtctl **start**, **stop** et **restart** déterminent l'utilisation de **RunStrategy**.

Le tableau suivant retrace la transition d'une VM entre différents états. La première colonne indique l'état initial de la VM : **RunStrategy**. Chaque colonne supplémentaire montre une commande virtctl et le nouveau site **RunStrategy** après l'exécution de cette commande.

Stratégie d'exécution initiale	commencer	arrêter	redémarrer
Always	-	Arrêté	Always
RerunOnFailure	-	Arrêté	RerunOnFailure
Manuel	Manuel	Manuel	Manuel
Arrêté	Always	-	-



NOTE

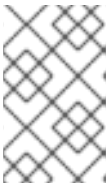
Dans les clusters OpenShift Virtualization installés à l'aide d'une infrastructure fournie par l'installateur, lorsqu'un nœud échoue au MachineHealthCheck et devient indisponible pour le cluster, les VM dont la stratégie d'exécution est **Always** ou **RerunOnFailure** sont replanifiées sur un nouveau nœud.

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  RunStrategy: Always (1)
  template:
...
```

1 Le paramètre actuel de l'IMV **RunStrategy**.

10.1.7. Ressources supplémentaires

- La définition de **VirtualMachineSpec** dans la référence API de KubeVirt v0.58.0 fournit un contexte plus large pour les paramètres et la hiérarchie de la spécification de la machine virtuelle.



NOTE

La référence de l'API KubeVirt est la référence du projet en amont et peut contenir des paramètres qui ne sont pas pris en charge dans OpenShift Virtualization.

- Activez le gestionnaire de CPU pour qu'il utilise le profil de charge de travail haute performance.
- Voir Préparer un disque de conteneur avant de l'ajouter à une machine virtuelle en tant que volume **containerDisk**.
- Pour plus d'informations sur le déploiement et l'activation des contrôles de santé des machines, voir Déploiement des contrôles de santé des machines.
- Pour plus de détails sur l'infrastructure fournie par l'installateur, voir Aperçu de l'infrastructure fournie par l' installateur.

- Personnalisation du profil de stockage

10.2. MODIFIER LES MACHINES VIRTUELLES

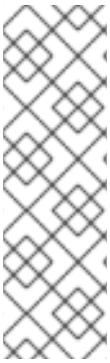
Vous pouvez mettre à jour la configuration d'une machine virtuelle en utilisant soit l'éditeur YAML dans la console web, soit l'OpenShift CLI sur la ligne de commande. Vous pouvez également mettre à jour un sous-ensemble de paramètres dans l'écran **Virtual Machine Details**.

10.2.1. Modifier une machine virtuelle dans la console web

Vous pouvez modifier une machine virtuelle en utilisant la console web d'OpenShift Container Platform ou l'interface de ligne de commande.

Procédure

1. Naviguez vers **Virtualization → VirtualMachines** dans la console web.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur n'importe quel champ comportant l'icône d'un crayon, ce qui indique que le champ est modifiable. Par exemple, cliquez sur le paramètre actuel de **Boot mode**, tel que BIOS ou UEFI, pour ouvrir la fenêtre **Boot mode** et sélectionner une option dans la liste.
4. Cliquez sur **Save**.



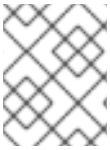
NOTE

Si la machine virtuelle est en cours d'exécution, les modifications apportées à **Boot Order** ou **Flavor** ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle.

Vous pouvez voir les changements en attente en cliquant sur **View Pending Changes** sur le côté droit du champ concerné. La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

10.2.2. Modifier la configuration YAML d'une machine virtuelle à l'aide de la console web

Vous pouvez éditer la configuration YAML d'une machine virtuelle dans la console web. Certains paramètres ne peuvent pas être modifiés. Si vous cliquez sur **Save** avec une configuration non valide, un message d'erreur indique le paramètre qui ne peut pas être modifié.



NOTE

Le fait de quitter l'écran YAML en cours d'édition annule toutes les modifications apportées à la configuration.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle.

3. Cliquez sur l'onglet **YAML** pour afficher la configuration modifiable.
4. Facultatif : vous pouvez cliquer sur **Download** pour télécharger localement le fichier YAML dans son état actuel.
5. Modifiez le fichier et cliquez sur **Save**.

Un message de confirmation indique que la modification a été effectuée avec succès et inclut le numéro de version mis à jour de l'objet.

10.2.3. Modifier la configuration YAML d'une machine virtuelle à l'aide du CLI

Utilisez cette procédure pour modifier la configuration YAML d'une machine virtuelle à l'aide de l'interface de programmation.

Conditions préalables

- Vous avez configuré une machine virtuelle avec un fichier de configuration d'objets YAML.
- Vous avez installé le CLI **oc**.

Procédure

1. Exécutez la commande suivante pour mettre à jour la configuration de la machine virtuelle :

```
oc edit <object_type> <object_ID> $ oc edit <object_type> <object_ID>
```

2. Ouvrez la configuration de l'objet.
3. Modifier le YAML.
4. Si vous modifiez une machine virtuelle en cours d'exécution, vous devez effectuer l'une des opérations suivantes :
   - Redémarrer la machine virtuelle.
   - Exécutez la commande suivante pour que la nouvelle configuration prenne effet :

```
oc apply <object_type> <object_ID> $ oc apply <object_type> <object_ID>
```

10.2.4. Ajouter un disque virtuel à une machine virtuelle

Cette procédure permet d'ajouter un disque virtuel à une machine virtuelle.

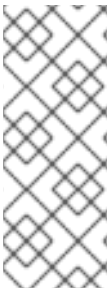
Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir l'écran **VirtualMachine details**.
3. Cliquez sur l'onglet **Disks** puis sur **Add disk**.
4. Dans la fenêtre **Add disk**, indiquez les adresses **Source**, **Name**, **Size**, **Type**, **Interface**, et **Storage Class**.

a. Facultatif : vous pouvez activer la pré-allocation si vous utilisez une source de disque vierge et que vous avez besoin de performances d'écriture maximales lors de la création de volumes de données. Pour ce faire, cochez la case **Enable preallocation**.

b. Facultatif : vous pouvez effacer **Apply optimized StorageProfile settings** pour modifier **Volume Mode** et **Access Mode** pour le disque virtuel. Si vous ne spécifiez pas ces paramètres, le système utilise les valeurs par défaut de la carte de configuration **kubevirt-storage-class-defaults**.

5. Cliquez sur **Add**.



NOTE

Si la machine virtuelle est en cours d'exécution, le nouveau disque est dans l'état **pending restart** et ne sera pas attaché tant que vous n'aurez pas redémarré la machine virtuelle.

La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

Pour configurer les valeurs par défaut des classes de stockage, utilisez les profils de stockage. Pour plus d'informations, voir Personnaliser le profil de stockage .

10.2.4.1. Édition de CD-ROM pour VirtualMachines

La procédure suivante permet d'éditer des CD-ROM pour les machines virtuelles.

Procédure

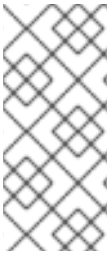
1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir l'écran **VirtualMachine details**.
3. Cliquez sur l'onglet **Disks**.
4. Cliquez sur le menu Options  du CD-ROM que vous souhaitez éditer et sélectionnez **Edit**.
5. Dans la fenêtre **Edit CD-ROM**, modifiez les champs : **Source**, **Persistent Volume Claim**, **Name**, **Type**, et **Interface**.
6. Cliquez sur **Save**.

10.2.4.2. Champs de stockage

Nom	La sélection	Description
Source	Vierge (crée du PVC)	Créer un disque vide.
	Importation par URL (crée un PVC)	Importer du contenu via une URL (HTTP ou HTTPS).

Nom	La sélection	Description
	Utiliser un PVC existant	Utiliser un PVC déjà disponible dans le cluster.
	Cloner un PVC existant (crée un PVC)	Sélectionnez un PVC existant disponible dans le cluster et clonez-le.
	Importation via le registre (création de PVC)	Importer du contenu via le registre des conteneurs.
	Conteneur (éphémère)	Télécharger le contenu d'un conteneur situé dans un registre accessible depuis le cluster. Le disque conteneur ne doit être utilisé que pour les systèmes de fichiers en lecture seule, tels que les CD-ROM ou les machines virtuelles temporaires.
Nom		Nom du disque. Le nom peut contenir des lettres minuscules (**a-z**), des chiffres (**0-9**), des traits d'union (**-**) et des points (**.**), jusqu'à un maximum de 253 caractères. Le premier et le dernier caractères doivent être alphanumériques. Le nom ne doit pas contenir de majuscules, d'espaces ou de caractères spéciaux.
Taille		Taille du disque en gigaoctets.
Type		Type de disque. Exemple : Disque ou CD-ROM
Interface		Type de périphérique de disque. Les interfaces prises en charge sont **virtIO**, **SATA** et **SCSI**.
Classe de stockage		La classe de stockage utilisée pour créer le disque.

Paramètres de stockage avancés
Les paramètres de stockage avancés suivants sont facultatifs et disponibles pour les disques **Blank**, **Import via URL**, et **Clone existing PVC**. Avant OpenShift Virtualization 4.11, si vous ne spécifiez pas ces paramètres, le système utilise les valeurs par défaut de la carte de configuration **kubevirt-storage-class-defaults**. Dans OpenShift Virtualization 4.11 et les versions ultérieures, le système utilise les valeurs par défaut du profil de stockage .



NOTE

Utilisez les profils de stockage pour garantir des paramètres de stockage avancés cohérents lors du provisionnement du stockage pour OpenShift Virtualization.

Pour spécifier manuellement **Volume Mode** et **Access Mode**, vous devez décocher la case **Apply optimized StorageProfile settings**, qui est sélectionnée par défaut.

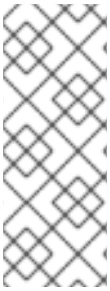
Nom	Description du mode	Paramètres	Description des paramètres
Mode volume	Définit si le volume persistant utilise un système de fichiers formaté ou un état de bloc brut. La valeur par défaut est **Filesystem**.	Système de fichiers	Stocke le disque virtuel sur un volume basé sur un système de fichiers.
		Bloc	Enregistre le disque virtuel directement sur le volume de blocs. N'utilisez **Block** que si le stockage sous-jacent le prend en charge.
Mode d'accès	Mode d'accès au volume persistant.	ReadWriteOnce (RWO)	Le volume peut être monté en lecture-écriture par un seul nœud.
		ReadWriteMany (RWX)	Le volume peut être monté en lecture-écriture par plusieurs nœuds à la fois.  NOTE Cela est nécessaire pour certaines fonctionnalités, telles que la migration en direct des machines virtuelles entre les nœuds.
		ReadOnlyMany (ROX)	Le volume peut être monté en lecture seule par plusieurs nœuds.

10.2.5. Ajouter une interface réseau à une machine virtuelle

Cette procédure permet d'ajouter une interface réseau à une machine virtuelle.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir l'écran **VirtualMachine details**.
3. Cliquez sur l'onglet **Network Interfaces**.
4. Cliquez sur **Add Network Interface**.
5. Dans la fenêtre **Add Network Interface**, indiquez les adresses **Name**, **Model**, **Network**, **Type** et **MAC Address** de l'interface réseau.
6. Cliquez sur **Add**.



NOTE

Si la machine virtuelle est en cours d'exécution, la nouvelle interface réseau est dans l'état **pending restart** et les changements ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle.

La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

10.2.5.1. Domaines de mise en réseau

Nom	Description
Nom	Nom du contrôleur d'interface réseau.
Model	Indique le modèle du contrôleur d'interface réseau. Les valeurs prises en charge sont **e1000e** et **virtio**.
Réseau	Liste des définitions de pièces jointes disponibles.
Type	Liste des méthodes de liaison disponibles. Sélectionnez la méthode de liaison adaptée à l'interface réseau : • Réseau de pods par défaut : **masquerade** • Réseau de ponts Linux : **bridge** • Réseau SR-IOV : **SR-IOV**
Adresse MAC	Adresse MAC du contrôleur d'interface réseau. Si aucune adresse MAC n'est spécifiée, une adresse est attribuée automatiquement.

10.2.6. Ressources supplémentaires

- Personnalisation du profil de stockage

10.3. MODIFICATION DE L'ORDRE DE DÉMARRAGE

Vous pouvez mettre à jour les valeurs d'une liste d'ordre de démarrage à l'aide de la console Web ou de l'interface de ligne de commande.

Avec **Boot Order** dans la page **Virtual Machine Overview**, vous pouvez :

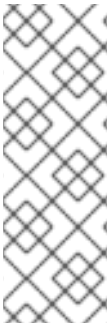
- Sélectionnez un disque ou un contrôleur d'interface réseau (NIC) et ajoutez-le à la liste de l'ordre de démarrage.
- Modifier l'ordre des disques ou des cartes d'interface réseau dans la liste de l'ordre de démarrage.
- Retirer un disque ou une carte d'interface réseau de la liste d'amorçage et le réintégrer dans l'inventaire des sources d'amorçage.

10.3.1. Ajouter des éléments à une liste de commandes de démarrage dans la console web

Ajouter des éléments à une liste de commandes de démarrage à l'aide de la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Details**.
4. Cliquez sur l'icône en forme de crayon située à droite de **Boot Order**. Si une configuration YAML n'existe pas, ou si c'est la première fois que vous créez une liste d'ordres de démarrage, le message suivant s'affiche : **No resource selected. VM will attempt to boot from disks by order of appearance in YAML file.**
5. Cliquez sur **Add Source** et sélectionnez un disque de démarrage ou un contrôleur d'interface réseau (NIC) pour la machine virtuelle.
6. Ajoutez tout disque ou carte d'interface réseau supplémentaire à la liste de l'ordre de démarrage.
7. Cliquez sur **Save**.



NOTE

Si la machine virtuelle est en cours d'exécution, les modifications apportées à **Boot Order** ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle.

Vous pouvez voir les changements en attente en cliquant sur **View Pending Changes** sur le côté droit du champ **Boot Order**. La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

10.3.2. Modification d'une liste de commandes de démarrage dans la console web

Modifiez la liste de l'ordre de démarrage dans la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Details**.
4. Cliquez sur l'icône du crayon qui se trouve sur le côté droit de **Boot Order**.
5. Choisissez la méthode appropriée pour déplacer l'élément dans la liste de l'ordre de démarrage :
   - Si vous n'utilisez pas de lecteur d'écran, survolez l'icône de la flèche située à côté de l'élément que vous souhaitez déplacer, faites glisser l'élément vers le haut ou vers le bas et déposez-le à l'endroit de votre choix.
   - Si vous utilisez un lecteur d'écran, appuyez sur la flèche du haut ou la flèche du bas pour déplacer l'élément dans la liste de l'ordre de démarrage. Ensuite, appuyez sur la touche **Tab** pour déposer l'élément à l'emplacement de votre choix.
6. Cliquez sur **Save**.

NOTE

Si la machine virtuelle est en cours d'exécution, les modifications apportées à la liste d'ordre de démarrage ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle.

Vous pouvez voir les changements en attente en cliquant sur **View Pending Changes** sur le côté droit du champ **Boot Order**. La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

10.3.3. Modification d'une liste d'ordre de démarrage dans le fichier de configuration YAML

Modifier la liste de l'ordre de démarrage dans un fichier de configuration YAML à l'aide de l'interface de programmation.

Procédure

1. Ouvrez le fichier de configuration YAML de la machine virtuelle en exécutant la commande suivante :

```
$ oc edit vm example
```

2. Editez le fichier YAML et modifiez les valeurs de l'ordre de démarrage associé à un disque ou à un contrôleur d'interface réseau (NIC). Par exemple :

```
disks:
  - bootOrder: 1 (1)
    disk:
      bus: virtio
    name: containerdisk
  - disk:
      bus: virtio
    name: cloudinitdisk
```

```
    - cdrom:
        bus: virtio
      name: cd-drive-1
  interfaces:
    - boot Order: 2 2
      macAddress: '02:96:c4:00:00'
      masquerade: {}
      name: default
```

1 Valeur de l'ordre de démarrage spécifiée pour le disque.

2 Valeur de l'ordre de démarrage spécifiée pour le contrôleur d'interface réseau.

3. Enregistrer le fichier YAML.
4. Cliquez sur **reload the content** pour appliquer les valeurs de l'ordre de démarrage mises à jour dans le fichier YAML à la liste de l'ordre de démarrage dans la console Web.

10.3.4. Suppression d'éléments d'une liste de commandes de démarrage dans la console web

Supprimer des éléments d'une liste de commandes de démarrage à l'aide de la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Details**.
4. Cliquez sur l'icône du crayon qui se trouve sur le côté droit de **Boot Order**.
5. Cliquez sur l'icône **Remove** à côté de l'élément. L'élément est supprimé de la liste d'amorçage et enregistré dans la liste des sources d'amorçage disponibles. Si vous supprimez tous les éléments de la liste d'amorçage, le message suivant s'affiche : **No resource selected. VM will attempt to boot from disks by order of appearance in YAML file.**

NOTE

Si la machine virtuelle est en cours d'exécution, les modifications apportées à **Boot Order** ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle.

Vous pouvez voir les changements en attente en cliquant sur **View Pending Changes** sur le côté droit du champ **Boot Order**. La bannière **Pending Changes** en haut de la page affiche une liste de tous les changements qui seront appliqués lorsque la machine virtuelle redémarrera.

10.4. SUPPRESSION DES MACHINES VIRTUELLES

Vous pouvez supprimer une machine virtuelle à partir de la console Web ou en utilisant l'interface de ligne de commande **oc**.

10.4.1. Suppression d'une machine virtuelle à l'aide de la console web

La suppression d'une machine virtuelle la retire définitivement du cluster.

NOTE

Lorsque vous supprimez une machine virtuelle, le volume de données qu'elle utilise est automatiquement supprimé.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Cliquez sur le menu Options  de la machine virtuelle que vous souhaitez supprimer et sélectionnez **Delete**.
   - Vous pouvez également cliquer sur le nom de la machine virtuelle pour ouvrir la page **VirtualMachine details** et cliquer sur **Actions → Delete**.
3. Dans la fenêtre de confirmation, cliquez sur **Delete** pour supprimer définitivement la machine virtuelle.

10.4.2. Suppression d'une machine virtuelle à l'aide du CLI

Vous pouvez supprimer une machine virtuelle en utilisant l'interface de ligne de commande (CLI) **oc**. Le client **oc** vous permet d'effectuer des actions sur plusieurs machines virtuelles.

NOTE

Lorsque vous supprimez une machine virtuelle, le volume de données qu'elle utilise est automatiquement supprimé.

Conditions préalables

- Identifiez le nom de la machine virtuelle que vous souhaitez supprimer.

Procédure

- Supprimez la machine virtuelle en exécutant la commande suivante :

```
oc delete vm <vm_name>
```

NOTE

Cette commande ne supprime que les objets qui existent dans le projet actuel. Spécifiez l'option **-n <project_name>** si l'objet que vous souhaitez supprimer se trouve dans un projet ou un espace de noms différent.

10.5. EXPORTER DES MACHINES VIRTUELLES

Vous pouvez exporter une machine virtuelle (VM) et ses disques associés afin d'importer une VM dans un autre cluster ou d'analyser le volume à des fins médico-légales.

Vous créez une ressource personnalisée (CR) **VirtualMachineExport** à l'aide de l'interface de ligne de commande.

Vous pouvez également utiliser la commande**virtctl vmexport** pour créer un CR **VirtualMachineExport** et télécharger les volumes exportés.

10.5.1. Création d'une ressource personnalisée VirtualMachineExport

Vous pouvez créer une ressource personnalisée (CR) **VirtualMachineExport** pour exporter les objets suivants :

- Machine virtuelle (VM) : Exporte les réclamations de volumes persistants (PVC) d'une VM spécifiée.
- Instantané de VM : Exporte les PVC contenus dans un CR **VirtualMachineSnapshot**.
- PVC : Exporte un PVC. Si le PVC est utilisé par un autre pod, tel que le pod **virt-launcher**, l'exportation reste dans l'état **Pending** jusqu'à ce que le PVC ne soit plus utilisé.

Le CR **VirtualMachineExport** crée des liens internes et externes pour les volumes exportés. Les liens internes sont valables au sein du cluster. Les liens externes sont accessibles à l'aide d'un CR **Ingress** ou **Route**.

Le serveur d'exportation prend en charge les formats de fichiers suivants :

- **raw**: Fichier image disque brut.
- **gzip**: Fichier image disque compressé.
- **dir**: Répertoire et fichiers PVC.
- **tar.gz**: Fichier PVC compressé.

Conditions préalables

- La machine virtuelle doit être arrêtée pour l'exportation de la machine virtuelle.

Procédure

1. Créez un manifeste **VirtualMachineExport** pour exporter un volume à partir d'un CR **VirtualMachine**, **VirtualMachineSnapshot** ou **PersistentVolumeClaim** selon l'exemple suivant et enregistrez-le sous **example-export.yaml**:

   VirtualMachineExport exemple

```
apiVersion: export.kubevirt.io/v1alpha1
kind: VirtualMachineExport
metadata:
  name: example-export
spec:
  source:
    apiGroup: "kubevirt.io" 1
```

```
  kind: VirtualMachine (2)
  name: example-vm
ttlDuration: 1h (3)
```

(1) Spécifiez le groupe API approprié :

- **"kubevirt.io"** pour **VirtualMachine**.
- **"snapshot.kubevirt.io"** pour **VirtualMachineSnapshot**.
- **""** pour **PersistentVolumeClaim**.

(2) Spécifiez **VirtualMachine**, **VirtualMachineSnapshot**, ou **PersistentVolumeClaim**.

(3) Facultatif. La durée par défaut est de 2 heures.

2. Créer le CR **VirtualMachineExport**:

```
$ oc create -f example-export.yaml
```

3. Obtenez le CR **VirtualMachineExport**:

```
$ oc get vmexport example-export -o yaml
```

Les liens internes et externes des volumes exportés sont affichés dans la strophe **status**:

Exemple de sortie

```
apiVersion: export.kubevirt.io/v1alpha1
kind: VirtualMachineExport
metadata:
  name: example-export
  namespace: example
spec:
  source:
    apiGroup: ""
    kind: PersistentVolumeClaim
    name: example-pvc
  tokenSecretRef: example-token
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2022-06-21T14:10:09Z"
    reason: podReady
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: "2022-06-21T14:09:02Z"
    reason: pvcBound
    status: "True"
    type: PVCReady
  links:
    external: (1)
      cert: |-
```

```
        -----BEGIN CERTIFICATE-----
        ...
        -----END CERTIFICATE-----
      volumes:
      - formats:
        - format: raw
          url: https://vmexport-
proxy.test.net/api/export.kubevirt.io/v1alpha1/namespaces/example/virtualmachineexports/exam
ple-export/volumes/example-disk/disk.img
        - format: gzip
          url: https://vmexport-
proxy.test.net/api/export.kubevirt.io/v1alpha1/namespaces/example/virtualmachineexports/exam
ple-export/volumes/example-disk/disk.img.gz
        name: example-disk
    internal:  2
      cert: |-
        -----BEGIN CERTIFICATE-----
        ...
        -----END CERTIFICATE-----
      volumes:
      - formats:
        - format: raw
          url: https://virt-export-example-export.example.svc/volumes/example-disk/disk.img
        - format: gzip
          url: https://virt-export-example-export.example.svc/volumes/example-disk/disk.img.gz
        name: example-disk
  phase: Ready
  serviceName: virt-export-example-export
```

1 Les liens externes sont accessibles depuis l'extérieur du cluster à l'aide d'un **Ingress** ou d'un **Route**.

2 Les liens internes ne sont valables qu'à l'intérieur du cluster.

10.6. GESTION DES INSTANCES DE MACHINES VIRTUELLES

Si vous avez des instances de machines virtuelles autonomes (VMI) qui ont été créées indépendamment en dehors de l'environnement OpenShift Virtualization, vous pouvez les gérer en utilisant la console web ou en utilisant les commandes **oc** ou **virtctl** à partir de l'interface de ligne de commande (CLI).

La commande **virtctl** offre plus d'options de virtualisation que la commande **oc**. Par exemple, vous pouvez utiliser **virtctl** pour mettre en pause une VM ou exposer un port.

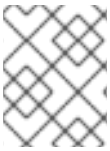
10.6.1. À propos des instances de machines virtuelles

Une instance de machine virtuelle (IMV) est une représentation d'une machine virtuelle (VM) en cours d'exécution. Lorsqu'une IMV appartient à une VM ou à un autre objet, vous la gérez par l'intermédiaire de son propriétaire dans la console Web ou à l'aide de l'interface de ligne de commande (CLI) **oc**.

Un VMI autonome est créé et démarré indépendamment avec un script, par automatisation, ou en utilisant d'autres méthodes dans le CLI. Dans votre environnement, vous pouvez avoir des IMV autonomes qui ont été développées et démarrées en dehors de l'environnement OpenShift Virtualization. Vous pouvez continuer à gérer ces IMV autonomes en utilisant le CLI. Vous pouvez également utiliser la console web pour des tâches spécifiques associées aux IMV autonomes :

- Dresser la liste des IMV autonomes et de leurs coordonnées.
- Modifier les étiquettes et les annotations pour une IMV autonome.
- Supprimer une VMI autonome.

Lorsque vous supprimez une VM, la VMI associée est automatiquement supprimée. Vous supprimez directement une VMI autonome car elle n'appartient pas à des VM ou à d'autres objets.



NOTE

Avant de désinstaller OpenShift Virtualization, listez et visualisez les VMIs autonomes en utilisant le CLI ou la console web. Ensuite, supprimez tous les VMI en suspens.

10.6.2. Liste de toutes les instances de machines virtuelles à l'aide de l'interface de gestion

Vous pouvez lister toutes les instances de machines virtuelles (IMV) de votre cluster, y compris les IMV autonomes et celles appartenant à des machines virtuelles, en utilisant l'interface de ligne de commande (CLI) **oc**.

Procédure

- Dressez la liste de toutes les IMV en exécutant la commande suivante :

```
$ oc get vmis
```

10.6.3. Lister les instances de machines virtuelles autonomes à l'aide de la console web

La console web vous permet de répertorier et de visualiser les instances de machines virtuelles (VMI) autonomes de votre cluster qui n'appartiennent pas à des machines virtuelles (VM).

NOTE

Les IMV appartenant à des machines virtuelles ou à d'autres objets ne sont pas affichées dans la console Web. La console Web n'affiche que les IMV autonomes. Si vous souhaitez répertorier toutes les IMV de votre cluster, vous devez utiliser le CLI.

Procédure

- Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
  Vous pouvez identifier un IMV autonome par un badge de couleur foncée à côté de son nom.

10.6.4. Modifier une instance de machine virtuelle autonome à l'aide de la console web

Vous pouvez modifier les annotations et les étiquettes d'une instance de machine virtuelle autonome (VMI) à l'aide de la console Web. Les autres champs ne sont pas modifiables.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez un IMV autonome pour ouvrir la page **VirtualMachineInstance details**.
3. Dans l'onglet **Details**, cliquez sur l'icône représentant un crayon à côté de **Annotations** ou **Labels**.
4. Apportez les modifications nécessaires et cliquez sur **Save**.

10.6.5. Suppression d'une instance de machine virtuelle autonome à l'aide du CLI

Vous pouvez supprimer une instance de machine virtuelle autonome (VMI) en utilisant l'interface de ligne de commande (CLI) **oc**.

Conditions préalables

- Identifiez le nom de la VMI que vous souhaitez supprimer.

Procédure

- Supprimez la VMI en exécutant la commande suivante :

```
oc delete vmi <vmi_name>
```

10.6.6. Suppression d'une instance de machine virtuelle autonome à l'aide de la console web

Supprimer une instance de machine virtuelle autonome (VMI) à partir de la console web.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Cliquez sur **Actions → Delete VirtualMachineInstance**.
3. Dans la fenêtre de confirmation, cliquez sur **Delete** pour supprimer définitivement l'IMV autonome.

10.7. CONTRÔLER LES ÉTATS DES MACHINES VIRTUELLES

Vous pouvez arrêter, démarrer, redémarrer et annuler la pause des machines virtuelles à partir de la console web.

Vous pouvez utiliser **virtctl** pour gérer les états des machines virtuelles et effectuer d'autres actions à partir de l'interface de gestion. Par exemple, vous pouvez utiliser **virtctl** pour forcer l'arrêt d'une machine virtuelle ou exposer un port.

10.7.1. Démarrer une machine virtuelle

Vous pouvez démarrer une machine virtuelle à partir de la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Recherchez la ligne qui contient la machine virtuelle que vous souhaitez démarrer.
3. Naviguez jusqu'au menu approprié pour votre cas d'utilisation :
   - Pour rester sur cette page, où vous pouvez effectuer des actions sur plusieurs machines virtuelles :
     a. Cliquez sur le menu Options  situé à l'extrémité droite de la rangée.
   - Pour afficher des informations complètes sur la machine virtuelle sélectionnée avant de la démarrer :
     a. Accédez à la page **VirtualMachine details** en cliquant sur le nom de la machine virtuelle.
     b. Cliquez sur **Actions**.
4. Sélectionnez **Restart**.
5. Dans la fenêtre de confirmation, cliquez sur **Start** pour démarrer la machine virtuelle.

NOTE

Lorsque vous démarrez une machine virtuelle provisionnée à partir d'une source **URL** pour la première fois, la machine virtuelle a un statut de **Importing** pendant qu'OpenShift Virtualization importe le conteneur à partir du point de terminaison URL. En fonction de la taille de l'image, ce processus peut prendre plusieurs minutes.

10.7.2. Redémarrer une machine virtuelle

Vous pouvez redémarrer une machine virtuelle en cours d'exécution à partir de la console web.

IMPORTANT

Pour éviter les erreurs, ne redémarrez pas une machine virtuelle lorsque son statut est **Importing**.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Recherchez la ligne qui contient la machine virtuelle que vous souhaitez redémarrer.
3. Naviguez jusqu'au menu approprié pour votre cas d'utilisation :
   - Pour rester sur cette page, où vous pouvez effectuer des actions sur plusieurs machines virtuelles :
     a. Cliquez sur le menu Options  situé à l'extrémité droite de la rangée.

- Pour afficher des informations complètes sur la machine virtuelle sélectionnée avant de la redémarrer :
    a. Accédez à la page **VirtualMachine details** en cliquant sur le nom de la machine virtuelle.
    b. Cliquez sur **Actions → Restart**.

4. Dans la fenêtre de confirmation, cliquez sur **Restart** pour redémarrer la machine virtuelle.

10.7.3. Arrêt d'une machine virtuelle

Vous pouvez arrêter une machine virtuelle à partir de la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Recherchez la ligne qui contient la machine virtuelle que vous souhaitez arrêter.
3. Naviguez jusqu'au menu approprié pour votre cas d'utilisation :
    - Pour rester sur cette page, où vous pouvez effectuer des actions sur plusieurs machines virtuelles :
        a. Cliquez sur le menu Options  situé à l'extrémité droite de la rangée.
    - Pour afficher des informations complètes sur la machine virtuelle sélectionnée avant de l'arrêter :
        a. Accédez à la page **VirtualMachine details** en cliquant sur le nom de la machine virtuelle.
        b. Cliquez sur **Actions → Stop**.
4. Dans la fenêtre de confirmation, cliquez sur **Stop** pour arrêter la machine virtuelle.

10.7.4. Désactiver une machine virtuelle

Vous pouvez annuler la pause d'une machine virtuelle à partir de la console web.

Conditions préalables

- Au moins une de vos machines virtuelles doit avoir un statut de **Paused**.

NOTE

Vous pouvez mettre en pause les machines virtuelles en utilisant le client **virtctl**.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Recherchez la ligne qui contient la machine virtuelle que vous souhaitez désactiver.
3. Naviguez jusqu'au menu approprié pour votre cas d'utilisation :

- Pour rester sur cette page, où vous pouvez effectuer des actions sur plusieurs machines virtuelles :

  a. Dans la colonne **Status**, cliquez sur **Paused**.

- Pour afficher des informations complètes sur la machine virtuelle sélectionnée avant de l'interrompre :

  a. Accédez à la page **VirtualMachine details** en cliquant sur le nom de la machine virtuelle.

  b. Cliquez sur l'icône du crayon qui se trouve sur le côté droit de **Status**.

4. Dans la fenêtre de confirmation, cliquez sur **Unpause** pour annuler la pause de la machine virtuelle.

10.8. ACCÈS AUX CONSOLES DES MACHINES VIRTUELLES

OpenShift Virtualization fournit différentes consoles de machines virtuelles que vous pouvez utiliser pour accomplir différentes tâches du produit. Vous pouvez accéder à ces consoles via la console web d'OpenShift Container Platform et en utilisant des commandes CLI.

10.8.1. Accéder aux consoles des machines virtuelles dans la console web d'OpenShift Container Platform

Vous pouvez vous connecter aux machines virtuelles en utilisant la console série ou la console VNC dans la console web d'OpenShift Container Platform.

Vous pouvez vous connecter aux machines virtuelles Windows en utilisant la console de visualisation du bureau, qui utilise le protocole RDP (remote desktop protocol), dans la console web d'OpenShift Container Platform.

10.8.1.1. Connexion à la console série

Connectez-vous à la console série d'une machine virtuelle en cours d'exécution à partir de l'onglet **Console** sur la page **VirtualMachine details** de la console Web.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Console**. La console VNC s'ouvre par défaut.
4. Cliquez sur **Disconnect** pour vous assurer qu'une seule session de console est ouverte à la fois. Sinon, la session de console VNC reste active en arrière-plan.
5. Cliquez sur la liste déroulante **VNC Console** et sélectionnez **Serial Console**.
6. Cliquez sur **Disconnect** pour mettre fin à la session de la console.
7. Optionnel : Ouvrez la console série dans une fenêtre séparée en cliquant sur **Open Console in New Window**.

10.8.1.2. Connexion à la console VNC

Connectez-vous à la console VNC d'une machine virtuelle en cours d'exécution à partir de l'onglet **Console** sur la page **VirtualMachine details** de la console Web.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Console**. La console VNC s'ouvre par défaut.
4. Facultatif : Ouvrez la console VNC dans une fenêtre séparée en cliquant sur **Open Console in New Window**.
5. Facultatif : Envoyez les combinaisons de touches à la machine virtuelle en cliquant sur **Send Key**.
6. Cliquez en dehors de la fenêtre de la console, puis cliquez sur **Disconnect** pour mettre fin à la session.

10.8.1.3. Se connecter à une machine virtuelle Windows avec RDP

La console **Desktop viewer**, qui utilise le protocole RDP (Remote Desktop Protocol), offre une meilleure expérience de la console pour la connexion aux machines virtuelles Windows.

Pour se connecter à une machine virtuelle Windows avec RDP, téléchargez le fichier **console.rdp** pour la machine virtuelle à partir de l'onglet **Console** sur la page **VirtualMachine details** de la console web et fournissez-le à votre client RDP préféré.

Conditions préalables

- Une machine virtuelle Windows en cours d'exécution avec l'agent invité QEMU installé. Le site **qemu-guest-agent** est inclus dans les pilotes VirtIO.
- Un client RDP installé sur une machine du même réseau que la machine virtuelle Windows.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Cliquez sur une machine virtuelle Windows pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Console**.
4. Dans la liste des consoles, sélectionnez **Desktop viewer**.
5. Cliquez sur **Launch Remote Desktop** pour télécharger le fichier **console.rdp**.
6. Faites référence au fichier **console.rdp** dans votre client RDP préféré pour vous connecter à la machine virtuelle Windows.

10.8.1.4. Passer d'un affichage de machine virtuelle à l'autre

Si votre machine virtuelle Windows (VM) est équipée d'un vGPU, vous pouvez passer de l'affichage par défaut à l'affichage du vGPU à l'aide de la console web.

Conditions préalables

- Le dispositif médiatisé est configuré dans la ressource personnalisée **HyperConverged** et affecté à la VM.
- La VM est en cours d'exécution.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines**
2. Sélectionnez une machine virtuelle Windows pour ouvrir l'écran **Overview**.
3. Cliquez sur l'onglet **Console**.
4. Dans la liste des consoles, sélectionnez **VNC console**.
5. Choisissez la combinaison de touches appropriée dans la liste **Send Key**:
   a. Pour accéder à l'affichage VM par défaut, sélectionnez **Ctl Alt 1**.
   b. Pour accéder à l'affichage vGPU, sélectionnez **Ctl Alt 2**.

Ressources supplémentaires

- Configuration des dispositifs à médiation

10.8.1.5. Copier la commande SSH à l'aide de la console web

Copiez la commande pour vous connecter au terminal d'une machine virtuelle (VM) via SSH.

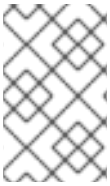
Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Cliquez sur le menu **Options**  de votre machine virtuelle et sélectionnez **Copy SSH command**.
3. Collez-le dans le terminal pour accéder à la VM.

10.8.2. Accès aux consoles des machines virtuelles à l'aide de commandes CLI

10.8.2.1. Accéder à une machine virtuelle via SSH en utilisant virtctl

Vous pouvez utiliser la commande **virtctl ssh** pour transférer le trafic SSH vers une machine virtuelle (VM) à l'aide de votre client SSH local.



NOTE

Un trafic SSH important sur le plan de contrôle peut ralentir le serveur API. Si vous avez régulièrement besoin d'un grand nombre de connexions, utilisez un objet Kubernetes **Service** dédié pour accéder à la machine virtuelle.

Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform avec les permissions **cluster-admin**.
- Vous avez installé l'OpenShift CLI (**oc**).
- Vous avez installé le client **virtctl**.
- La machine virtuelle à laquelle vous souhaitez accéder est en cours d'exécution.
- Vous êtes dans le même projet que la VM.

Procédure

1. Utilisez la commande **ssh-keygen** pour générer une paire de clés publiques SSH :

```
$ ssh-keygen -f <key_file> 1
```

1 Indiquez le fichier dans lequel les clés doivent être stockées.

2. Créez un secret d'authentification SSH qui contient la clé publique SSH pour accéder à la VM :

```
oc create secret generic my-pub-key --from-file=key1=<key_file>.pub
```

3. Ajoutez une référence au secret dans le manifeste **VirtualMachine**. Par exemple :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: testvm
spec:
  running: true
  template:
    spec:
      accessCredentials:
      - sshPublicKey:
          source:
            secret:
              secretName: my-pub-key 1
          propagationMethod:
            configDrive: {} 2
# ...
```

1 Référence à l'objet d'authentification SSH **Secret**.

2 La clé publique SSH est injectée dans la VM en tant que métadonnée de démarrage à l'aide du fournisseur **configDrive**.

4. Redémarrez la VM pour appliquer vos modifications.
5. Exécutez la commande suivante pour accéder à la VM via SSH :

```
virtctl ssh -i <key_file> <vusername>@<vm_name>
```

6. Facultatif : Pour transférer en toute sécurité des fichiers vers ou depuis la VM, utilisez les commandes suivantes :

Copier un fichier de votre machine vers la VM

```
virtctl scp -i <key_file> <filename> <vusername>@<vm_name> :
```

Copier un fichier de la VM vers votre machine

```
virtctl scp -i <key_file> <vusername@<vm_name>:<filename> .
```

Ressources supplémentaires

- Création d'un service pour exposer une machine virtuelle
- Comprendre les secrets

10.8.2.2. Accéder à la console série d'une instance de machine virtuelle

La commande **virtctl console** ouvre une console série sur l'instance de machine virtuelle spécifiée.

Conditions préalables

- Le paquet **virt-viewer** doit être installé.
- L'instance de machine virtuelle à laquelle vous souhaitez accéder doit être en cours d'exécution.

Procédure

- Connectez-vous à la console série avec **virtctl**:

```
$ virtctl console <VMI>
```

10.8.2.3. Accéder à la console graphique d'une instance de machine virtuelle avec VNC

L'utilitaire client **virtctl** peut utiliser la fonction **remote-viewer** pour ouvrir une console graphique sur une instance de machine virtuelle en cours d'exécution. Cette fonctionnalité est incluse dans le paquetage **virt-viewer**.

Conditions préalables

- Le paquet **virt-viewer** doit être installé.
- L'instance de machine virtuelle à laquelle vous souhaitez accéder doit être en cours d'exécution.

NOTE

Si vous utilisez **virtctl** via SSH sur une machine distante, vous devez transférer la session X vers votre machine.

Procédure

1. Connectez-vous à l'interface graphique avec l'utilitaire **virtctl**:

```
$ virtctl vnc <VMI>
```

2. Si la commande a échoué, essayez d'utiliser l'indicateur **-v** pour collecter des informations de dépannage :

```
$ virtctl vnc <VMI> -v 4
```

10.8.2.4. Se connecter à une machine virtuelle Windows avec une console RDP

Créez un objet Kubernetes **Service** pour vous connecter à une machine virtuelle (VM) Windows à l'aide de votre client RDP (Remote Desktop Protocol) local.

Conditions préalables

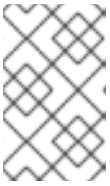
- Une machine virtuelle Windows en cours d'exécution avec l'agent invité QEMU installé. L'objet **qemu-guest-agent** est inclus dans les pilotes VirtIO.
- Un client RDP installé sur votre machine locale.

Procédure

1. Modifiez le manifeste **VirtualMachine** pour ajouter l'étiquette de création de service :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: vm-ephemeral
  namespace: example-namespace
spec:
  running: false
  template:
    metadata:
      labels:
        special: key 1
# ...
```

1 Ajouter le libellé **special: key** dans la section **spec.template.metadata.labels**.



NOTE

Les étiquettes d'une machine virtuelle sont transmises au pod. L'étiquette **special: key** doit correspondre à l'étiquette de l'attribut **spec.selector** du manifeste **Service**.

2. Enregistrez le fichier manifeste **VirtualMachine** pour appliquer vos modifications.

3. Créez un manifeste **Service** pour exposer la VM :

```
apiVersion: v1
kind: Service
metadata:
  name: rdpservice (1)
  namespace: example-namespace (2)
spec:
  ports:
  - targetPort: 3389 (3)
    protocol: TCP
  selector:
    special: key (4)
  type: NodePort (5)
# ...
```

(1) Le nom de l'objet **Service**.

(2) L'espace de noms dans lequel réside l'objet **Service**. Il doit correspondre au champ **metadata.namespace** du manifeste **VirtualMachine**.

(3) Le port VM à exposer par le service. Il doit faire référence à un port ouvert si une liste de ports est définie dans le manifeste VM.

(4) La référence à l'étiquette que vous avez ajoutée dans la strophe **spec.template.metadata.labels** du manifeste **VirtualMachine**.

(5) Le type de service.

4. Enregistrez le fichier manifeste **Service**.

5. Créez le service en exécutant la commande suivante :

```
oc create -f <service_name>.yaml
```

6. Démarrez la VM. Si la VM est déjà en cours d'exécution, redémarrez-la.

7. Interroger l'objet **Service** pour vérifier qu'il est disponible :

```
$ oc get service -n example-namespace
```

Exemple de sortie pour le service NodePort

```
NAME         TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
rdpservice   NodePort   172.30.232.73  <none>        3389:30000/TCP   5m
```

8. Exécutez la commande suivante pour obtenir l'adresse IP du nœud :

```
$ oc get node <node_name> -o wide
```

Exemple de sortie

```
NAME   STATUS  ROLES  AGE   VERSION  INTERNAL-IP     EXTERNAL-IP
node01 Ready   worker 6d22h v1.24.0 192.168.55.101  <none>
```

9. Spécifiez l'adresse IP du nœud et le port attribué dans votre client RDP préféré.

10. Saisissez le nom d'utilisateur et le mot de passe pour vous connecter à la machine virtuelle Windows.

10.9. AUTOMATISER L'INSTALLATION DE WINDOWS AVEC SYSPREP

Vous pouvez utiliser les images DVD Microsoft et **sysprep** pour automatiser l'installation, la configuration et l'approvisionnement en logiciels des machines virtuelles Windows.

10.9.1. Utilisation d'un DVD Windows pour créer une image disque de VM

Microsoft ne fournit pas d'images de disque à télécharger, mais vous pouvez créer une image de disque à l'aide d'un DVD Windows. Cette image disque peut ensuite être utilisée pour créer des machines virtuelles.

Procédure

1. Dans la console web d'OpenShift Virtualization, cliquez sur **Storage → PersistentVolumeClaims → Create PersistentVolumeClaim With Data upload form**

2. Sélectionnez le projet envisagé.

3. Régler l'adresse **Persistent Volume Claim Name**

4. Téléchargez l'image disque de la VM à partir du DVD Windows. L'image est maintenant disponible comme source de démarrage pour créer une nouvelle VM Windows.

10.9.2. Utilisation d'une image disque pour installer Windows

Vous pouvez utiliser une image disque pour installer Windows sur votre machine virtuelle.

Conditions préalables

- Vous devez créer une image disque à l'aide d'un DVD Windows.
- Vous devez créer un fichier de réponse **autounattend.xml**. Voir la documentation Microsoft pour plus de détails.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Catalog** dans le menu latéral.

2. Sélectionnez un modèle Windows et cliquez sur **Customize VirtualMachine**.

3. Sélectionnez **Upload (Upload a new file to a PVC)**dans la liste **Disk source** et naviguez jusqu'à l'image du DVD.

4. Cliquez sur **Review and create VirtualMachine**.

5. Clear **Clone available operating system source to this Virtual Machine**
6. Clear **Start this VirtualMachine after creation**.
7. Dans la section **Sysprep** de l'onglet **Scripts**, cliquez sur **Edit**.
8. Recherchez le fichier de réponse **autounattend.xml** et cliquez sur **Save**.
9. Cliquez sur **Create VirtualMachine**.
10. Dans l'onglet **YAML**, remplacez **running:false** par **runStrategy: RerunOnFailure** et cliquez sur **Save**.

La VM démarre avec le disque **sysprep** contenant le fichier de réponse **autounattend.xml**.

10.9.3. Généralisation d'une VM Windows à l'aide de sysprep

La généralisation d'une image permet à cette image de supprimer toutes les données de configuration spécifiques au système lorsque l'image est déployée sur une machine virtuelle (VM).

Avant de généraliser la VM, vous devez vous assurer que l'outil **sysprep** ne peut pas détecter un fichier de réponse après l'installation de Windows sans surveillance.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines**.
2. Sélectionnez une VM Windows pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Disks**.
4. Cliquez sur le menu Options  pour le disque **sysprep** et sélectionnez **Detach**.
5. Cliquez sur **Detach**.
6. Renommez **C:\Windows\Panther\unattend.xml** pour éviter qu'il ne soit détecté par l'outil **sysprep**.
7. Lancez le programme **sysprep** en exécutant la commande suivante :

```
%WINDIR%\System32\Sysprep\sysprep.exe /generalize /shutdown /oobe /mode:vm
```

8. Une fois l'outil **sysprep** terminé, la VM Windows s'arrête. L'image disque de la VM est maintenant disponible pour être utilisée comme image d'installation pour les VM Windows.

Vous pouvez maintenant spécialiser la VM.

10.9.4. Spécialisation d'une machine virtuelle Windows

La spécialisation d'une machine virtuelle (VM) permet de configurer les informations spécifiques à l'ordinateur à partir d'une image Windows généralisée sur la VM.

Conditions préalables

- Vous devez disposer d'une image disque Windows généralisée.
- Vous devez créer un fichier de réponse **unattend.xml**. Voir la documentation Microsoft pour plus de détails.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Catalog**.
2. Sélectionnez un modèle Windows et cliquez sur **Customize VirtualMachine**.
3. Sélectionnez **PVC (clone PVC)** dans la liste **Disk source**.
4. Spécifiez les adresses **Persistent Volume Claim project**et **Persistent Volume Claim name**de l'image Windows généralisée.
5. Cliquez sur **Review and create VirtualMachine**.
6. Cliquez sur l'onglet **Scripts**.
7. Dans la section **Sysprep**, cliquez sur **Edit**, recherchez le fichier réponse **unattend.xml** et cliquez sur **Save**.
8. Cliquez sur **Create VirtualMachine**.

Lors du démarrage initial, Windows utilise le fichier de réponse **unattend.xml** pour spécialiser la VM. La VM est maintenant prête à être utilisée.

10.9.5. Ressources supplémentaires

- Création de machines virtuelles
- Microsoft, Sysprep (Généraliser) une installation Windows
- Microsoft, généraliser
- Microsoft, se spécialiser

10.10. DÉCLENCHER LE BASCULEMENT D'UNE MACHINE VIRTUELLE EN RÉSOLVANT UN NŒUD DÉFAILLANT

Si un nœud tombe en panne et que les contrôles de santé des machines ne sont pas déployés sur votre cluster, les machines virtuelles (VM) configurées sur **RunStrategy: Always** ne sont pas automatiquement relocalisées sur des nœuds sains. Pour déclencher le basculement des machines virtuelles, vous devez supprimer manuellement l'objet **Node**.

NOTE

Si vous avez installé votre cluster en utilisant une infrastructure fournie par l'installateur et que vous avez correctement configuré les contrôles de santé des machines :

- Les nœuds défaillants sont automatiquement recyclés.
- Les machines virtuelles dont l'adresse **RunStrategy** ayant pour valeur **Always** ou **RerunOnFailure** sont automatiquement planifiées sur des nœuds sains.

10.10.1. Conditions préalables

- Un nœud où une machine virtuelle était en cours d'exécution présente la condition **NotReady**.
- La machine virtuelle qui s'exécutait sur le nœud défaillant a pour valeur **RunStrategy Always** .
- Vous avez installé l'OpenShift CLI (**oc**).

10.10.2. Suppression de nœuds d'un cluster bare metal

Lorsque vous supprimez un nœud à l'aide de la CLI, l'objet nœud est supprimé dans Kubernetes, mais les pods qui existent sur le nœud ne sont pas supprimés. Tous les pods nus qui ne sont pas soutenus par un contrôleur de réplication deviennent inaccessibles à OpenShift Container Platform. Les pods soutenus par des contrôleurs de réplication sont replanifiés sur d'autres nœuds disponibles. Vous devez supprimer les pods de manifeste locaux.

Procédure

Supprimez un nœud d'un cluster OpenShift Container Platform fonctionnant sur du métal nu en effectuant les étapes suivantes :

1. Marquer le nœud comme non ordonnançable :

   ```
   $ oc adm cordon <node_name>
   ```

2. Drainer tous les pods sur le nœud :

   ```
   oc adm drain <node_name> --force=true
   ```

   Cette étape peut échouer si le nœud est hors ligne ou ne répond pas. Même si le nœud ne répond pas, il est possible qu'il exécute toujours une charge de travail qui écrit dans le stockage partagé. Pour éviter toute corruption de données, mettez le matériel physique hors tension avant de poursuivre.

3. Supprimer le nœud de la grappe :

   ```
   oc delete node <node_name> $ oc delete node <node_name>
   ```

   Bien que l'objet nœud soit désormais supprimé du cluster, il peut toujours rejoindre le cluster après un redémarrage ou si le service kubelet est redémarré. Pour supprimer définitivement le nœud et toutes ses données, vous devez le déclasser.

4. Si vous avez mis le matériel physique hors tension, remettez-le sous tension pour que le nœud puisse rejoindre le cluster.

10.10.3. Vérification du basculement d'une machine virtuelle

Une fois que toutes les ressources ont été supprimées sur le nœud malsain, une nouvelle instance de machine virtuelle (VMI) est automatiquement créée sur un nœud sain pour chaque VM relocalisée. Pour confirmer que la VMI a été créée, affichez toutes les VMI à l'aide du CLI **oc**.

10.10.3.1. Liste de toutes les instances de machines virtuelles à l'aide de l'interface de gestion

Vous pouvez lister toutes les instances de machines virtuelles (IMV) de votre cluster, y compris les IMV autonomes et celles appartenant à des machines virtuelles, en utilisant l'interface de ligne de commande (CLI) **oc**.

Procédure

- Dressez la liste de toutes les IMV en exécutant la commande suivante :

```
$ oc get vmis
```

10.11. INSTALLATION DE L'AGENT INVITÉ QEMU SUR LES MACHINES VIRTUELLES

L'agent invité QEMU est un démon qui s'exécute sur la machine virtuelle et transmet à l'hôte des informations sur la machine virtuelle, les utilisateurs, les systèmes de fichiers et les réseaux secondaires.

10.11.1. Installation de l'agent invité QEMU sur une machine virtuelle Linux

Le site **qemu-guest-agent** est largement disponible et disponible par défaut dans les machines virtuelles Red Hat. Installez l'agent et démarrez le service.

Pour vérifier si l'agent invité QEMU est installé et fonctionne sur votre machine virtuelle (VM), vérifiez que **AgentConnected** figure dans les spécifications de la VM.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

Procédure

1. Accédez à la ligne de commande de la machine virtuelle via l'une des consoles ou via SSH.
2. Installer l'agent invité QEMU sur la machine virtuelle :

```
$ yum install -y qemu-guest-agent
```

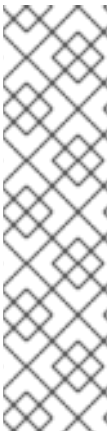
3. Assurez-vous que le service est persistant et démarrez-le :

```
$ systemctl enable --now qemu-guest-agent
```

10.11.2. Installation de l'agent invité QEMU sur une machine virtuelle Windows

Pour les machines virtuelles Windows, l'agent invité QEMU est inclus dans les pilotes VirtIO. Installer les pilotes sur une installation Windows existante ou nouvelle.

Pour vérifier si l'agent invité QEMU est installé et fonctionne sur votre machine virtuelle (VM), vérifiez que **AgentConnected** figure dans les spécifications de la VM.



NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

10.11.2.1. Installation des pilotes VirtIO sur une machine virtuelle Windows existante

Installer les pilotes VirtIO à partir du lecteur CD SATA connecté à une machine virtuelle Windows existante.

NOTE

Cette procédure utilise une approche générique pour ajouter des pilotes à Windows. La procédure peut différer légèrement d'une version de Windows à l'autre. Consultez la documentation d'installation de votre version de Windows pour connaître les étapes spécifiques de l'installation.

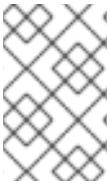
Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.
2. Se connecter à une session utilisateur Windows.
3. Ouvrez **Device Manager** et développez **Other devices** pour répertorier tous les **Unknown device**.
    a. Ouvrez le site **Device Properties** pour identifier l'appareil inconnu. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Properties**.
    b. Cliquez sur l'onglet **Details** et sélectionnez **Hardware Ids** dans la liste **Property**.
    c. Comparez le site **Value** pour le site **Hardware Ids** avec les pilotes VirtIO pris en charge.
4. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Update Driver Software**.
5. Cliquer sur **Browse my computer for driver software** et naviguer jusqu'au lecteur de CD SATA connecté, où se trouvent les pilotes VirtIO. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.
6. Cliquez sur **Next** pour installer le pilote.
7. Répéter ce processus pour tous les pilotes VirtIO nécessaires.
8. Après l'installation du pilote, cliquez sur **Close** pour fermer la fenêtre.

9. Redémarrez la machine virtuelle pour terminer l'installation du pilote.

10.11.2.2. Installation des pilotes VirtIO pendant l'installation de Windows

Installer les pilotes VirtIO à partir du pilote du CD SATA pendant l'installation de Windows.



NOTE

Cette procédure utilise une approche générique de l'installation de Windows et la méthode d'installation peut différer d'une version à l'autre de Windows. Consultez la documentation de la version de Windows que vous installez.

Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.
2. Commencez le processus d'installation de Windows.
3. Sélectionnez l'installation **Advanced**.
4. La destination de stockage ne sera pas reconnue tant que le pilote n'aura pas été chargé. Cliquez sur **Load driver**.
5. Les pilotes sont attachés à un lecteur de CD SATA. Cliquez sur **OK** et recherchez dans le lecteur de CD le pilote de stockage à charger. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.
6. Répétez les deux étapes précédentes pour tous les pilotes nécessaires.
7. Terminer l'installation de Windows.

10.12. AFFICHAGE DES INFORMATIONS RELATIVES À L'AGENT INVITÉ QEMU POUR LES MACHINES VIRTUELLES

Lorsque l'agent invité QEMU fonctionne sur la machine virtuelle, vous pouvez utiliser la console web pour afficher des informations sur la machine virtuelle, les utilisateurs, les systèmes de fichiers et les réseaux secondaires.

10.12.1. Conditions préalables

- Installer l'agent invité QEMU sur la machine virtuelle.

10.12.2. À propos des informations sur l'agent invité QEMU dans la console web

Lorsque l'agent invité QEMU est installé, les onglets **Overview** et **Details** de la page **VirtualMachine details** affichent des informations sur le nom d'hôte, le système d'exploitation, le fuseau horaire et les utilisateurs connectés.

La page **VirtualMachine details** affiche des informations sur le système d'exploitation invité installé sur la machine virtuelle. L'onglet **Details** affiche un tableau avec des informations sur les utilisateurs connectés. L'onglet **Disks** affiche un tableau avec des informations sur les systèmes de fichiers.

NOTE

Si l'agent invité QEMU n'est pas installé, les onglets **Overview** et **Details** affichent des informations sur le système d'exploitation spécifié lors de la création de la machine virtuelle.

10.12.3. Afficher les informations sur l'agent invité QEMU dans la console web

Vous pouvez utiliser la console web pour afficher les informations relatives aux machines virtuelles transmises par l'agent invité QEMU à l'hôte.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez un nom de machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Details** pour afficher les utilisateurs actifs.
4. Cliquez sur l'onglet **Disks** pour afficher des informations sur les systèmes de fichiers.

10.13. GESTION DES CARTES DE CONFIGURATION, DES SECRETS ET DES COMPTES DE SERVICE DANS LES MACHINES VIRTUELLES

Vous pouvez utiliser des secrets, des cartes de configuration et des comptes de service pour transmettre des données de configuration aux machines virtuelles. Par exemple, vous pouvez

- Donner à une machine virtuelle l'accès à un service nécessitant des informations d'identification en ajoutant un secret à la machine virtuelle.
- Stocker les données de configuration non confidentielles dans une carte de configuration afin qu'un pod ou un autre objet puisse consommer ces données.
- Permet à un composant d'accéder au serveur API en associant un compte de service à ce composant.

NOTE

OpenShift Virtualization expose les secrets, les cartes de configuration et les comptes de service en tant que disques de machine virtuelle afin que vous puissiez les utiliser sur toutes les plateformes sans surcharge supplémentaire.

10.13.1. Ajout d'un secret, d'une carte de configuration ou d'un compte de service à une machine virtuelle

Vous ajoutez un secret, une carte de configuration ou un compte de service à une machine virtuelle en utilisant la console web d'OpenShift Container Platform.

Ces ressources sont ajoutées à la machine virtuelle en tant que disques. Vous montez ensuite le secret, la carte de configuration ou le compte de service comme vous le feriez pour n'importe quel autre disque.

Si la machine virtuelle est en cours d'exécution, les changements ne prendront pas effet tant que vous n'aurez pas redémarré la machine virtuelle. Les ressources nouvellement ajoutées sont marquées comme étant en attente de modifications pour les onglets **Environment** et **Disks** dans la bannière

Pending Changes en haut de la page.

Conditions préalables

- Le secret, la carte de configuration ou le compte de service que vous souhaitez ajouter doit exister dans le même espace de noms que la machine virtuelle cible.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Dans l'onglet **Environment**, cliquez sur **Add Config Map, Secret or Service Account**
4. Cliquez sur **Select a resource** et sélectionnez une ressource dans la liste. Un numéro de série de six caractères est automatiquement généré pour la ressource sélectionnée.
5. Facultatif : cliquez sur pour rétablir l'environnement à son dernier état enregistré : Cliquez sur **Reload** pour rétablir le dernier état enregistré de l'environnement.
6. Cliquez sur **Save**.

Vérification

1. Sur la page **VirtualMachine details**, cliquez sur l'onglet **Disks** et vérifiez que le secret, la carte de configuration ou le compte de service est inclus dans la liste des disques.
2. Redémarrez la machine virtuelle en cliquant sur **Actions → Restart**.

Vous pouvez maintenant monter le secret, la carte de configuration ou le compte de service comme vous le feriez pour n'importe quel autre disque.

10.13.2. Suppression d'un secret, d'une carte de configuration ou d'un compte de service d'une machine virtuelle

Supprimer un secret, une carte de configuration ou un compte de service d'une machine virtuelle en utilisant la console web d'OpenShift Container Platform.

Conditions préalables

- Vous devez avoir au moins un secret, une carte de configuration ou un compte de service attaché à une machine virtuelle.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Environment**.
4. Recherchez l'élément que vous souhaitez supprimer dans la liste et cliquez sur **Remove** à droite de l'élément.

5. Cliquez sur **Save**.

NOTE

Vous pouvez rétablir le dernier état enregistré du formulaire en cliquant sur **Reload**.

Vérification

1. Sur la page **VirtualMachine details**, cliquez sur l'onglet **Disks**.
2. Vérifiez que le secret, la carte de configuration ou le compte de service que vous avez supprimé ne figure plus dans la liste des disques.

10.13.3. Ressources supplémentaires

- Fournir des données sensibles aux pods
- Comprendre et créer des comptes de service
- Comprendre les cartes de configuration

10.14. INSTALLATION DU PILOTE VIRTIO SUR UNE MACHINE VIRTUELLE WINDOWS EXISTANTE

10.14.1. À propos des pilotes VirtIO

Les pilotes VirtIO sont des pilotes de périphériques paravirtualisés requis pour que les machines virtuelles Microsoft Windows fonctionnent dans OpenShift Virtualization. Les pilotes pris en charge sont disponibles dans le disque de conteneur **container-native-virtualization/virtio-win** du catalogue de l'écosystème Red Hat.

Le disque conteneur **container-native-virtualization/virtio-win** doit être attaché à la machine virtuelle en tant que lecteur CD SATA pour permettre l'installation du pilote. Vous pouvez installer les pilotes VirtIO pendant l'installation de Windows sur la machine virtuelle ou ajoutés à une installation Windows existante.

Une fois les pilotes installés, le disque conteneur **container-native-virtualization/virtio-win** peut être retiré de la machine virtuelle.

Voir aussi : Installation des pilotes Virtio sur une nouvelle machine virtuelle Windows .

10.14.2. Pilotes VirtIO pris en charge pour les machines virtuelles Microsoft Windows

Tableau 10.1. Pilotes pris en charge

Nom du conducteur	ID du matériel	Description
viostor	VEN_1AF4&DEV_1001 VEN_1AF4&DEV_1042	Le pilote de bloc. Se présente parfois sous la forme d'un **SCSI Controller** dans le groupe **Other devices**.

Nom du conducteur	ID du matériel	Description
viorng	VEN_1AF4&DEV_1005 VEN_1AF4&DEV_1044	Le pilote de la source d'entropie. S'affiche parfois sous la forme d'un **PCI Device** dans le groupe **Other devices**.
NetKVM	VEN_1AF4&DEV_1000 VEN_1AF4&DEV_1041	Le pilote de réseau. S'affiche parfois comme **Ethernet Controller** dans le groupe **Other devices**. Disponible uniquement si un NIC VirtIO est configuré.

10.14.3. Ajout d'un disque conteneur de pilotes VirtIO à une machine virtuelle

OpenShift Virtualization distribue les pilotes VirtIO pour Microsoft Windows sous la forme d'un disque conteneur, qui est disponible à partir du catalogue de l'écosystème Red Hat . Pour installer ces pilotes sur une machine virtuelle Windows, attachez le disque conteneur **container-native-virtualization/virtio-win** à la machine virtuelle en tant que lecteur de CD SATA dans le fichier de configuration de la machine virtuelle.

Conditions préalables

- Téléchargez le disque de conteneur **container-native-virtualization/virtio-win** à partir du catalogue de l'écosystème Red Hat . Ceci n'est pas obligatoire, car le disque conteneur sera téléchargé depuis le registre Red Hat s'il n'est pas déjà présent dans le cluster, mais cela peut réduire le temps d'installation.

Procédure

1. Ajoutez le disque conteneur **container-native-virtualization/virtio-win** en tant que disque **cdrom** dans le fichier de configuration de la machine virtuelle Windows. Le disque conteneur sera téléchargé depuis le registre s'il n'est pas déjà présent dans le cluster.

```
spec:
  domain:
    devices:
      disks:
        - name: virtiocontainerdisk
          bootOrder: 2 (1)
          cdrom:
            bus: sata
volumes:
  - containerDisk:
      image: container-native-virtualization/virtio-win
    name: virtiocontainerdisk
```

(1) OpenShift Virtualization démarre les disques de la machine virtuelle dans l'ordre défini dans le fichier de configuration **VirtualMachine**. Vous pouvez soit définir d'autres disques pour la machine virtuelle avant le disque du conteneur **container-native-**

virtualization/virtio-win, soit utiliser le paramètre facultatif **bootOrder** pour vous assurer que la machine virtuelle démarre à partir du bon disque. Si vous spécifiez le paramètre **bootOrder** pour un disque, il doit être spécifié pour tous les disques de la configuration.

2. Le disque est disponible dès que la machine virtuelle a démarré :
   - Si vous ajoutez le disque conteneur à une machine virtuelle en cours d'exécution, utilisez **oc apply -f <vm.yaml>** dans l'interface CLI ou redémarrez la machine virtuelle pour que les modifications soient prises en compte.
   - Si la machine virtuelle n'est pas en cours d'exécution, utilisez **virtctl start <vm>**.

Après le démarrage de la machine virtuelle, les pilotes VirtIO peuvent être installés à partir du lecteur CD SATA connecté.

10.14.4. Installation des pilotes VirtIO sur une machine virtuelle Windows existante

Installer les pilotes VirtIO à partir du lecteur CD SATA connecté à une machine virtuelle Windows existante.

NOTE

Cette procédure utilise une approche générique pour ajouter des pilotes à Windows. La procédure peut différer légèrement d'une version de Windows à l'autre. Consultez la documentation d'installation de votre version de Windows pour connaître les étapes spécifiques de l'installation.

Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.
2. Se connecter à une session utilisateur Windows.
3. Ouvrez **Device Manager** et développez **Other devices** pour répertorier tous les **Unknown device**.
   a. Ouvrez le site **Device Properties** pour identifier l'appareil inconnu. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Properties**.
   b. Cliquez sur l'onglet **Details** et sélectionnez **Hardware Ids** dans la liste **Property**.
   c. Comparez le site **Value** pour le site **Hardware Ids** avec les pilotes VirtIO pris en charge.
4. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Update Driver Software**.
5. Cliquer sur **Browse my computer for driver software**et naviguer jusqu'au lecteur de CD SATA connecté, où se trouvent les pilotes VirtIO. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.
6. Cliquez sur **Next** pour installer le pilote.
7. Répéter ce processus pour tous les pilotes VirtIO nécessaires.
8. Après l'installation du pilote, cliquez sur **Close** pour fermer la fenêtre.
9. Redémarrez la machine virtuelle pour terminer l'installation du pilote.

10.14.5. Suppression du disque du conteneur VirtIO d'une machine virtuelle

Après avoir installé tous les pilotes VirtIO requis sur la machine virtuelle, le disque conteneur **container-native-virtualization/virtio-win** n'a plus besoin d'être attaché à la machine virtuelle. Supprimez le disque conteneur **container-native-virtualization/virtio-win** du fichier de configuration de la machine virtuelle.

Procédure

1. Modifiez le fichier de configuration et supprimez **disk** et **volume**.

```
$ oc edit vm <vm-name>
```

```
spec:
  domain:
    devices:
      disks:
        - name: virtiocontainerdisk
          bootOrder: 2
          cdrom:
            bus: sata
volumes:
  - containerDisk:
      image: container-native-virtualization/virtio-win
    name: virtiocontainerdisk
```

2. Redémarrez la machine virtuelle pour que les modifications soient prises en compte.

10.15. INSTALLATION DU PILOTE VIRTIO SUR UNE NOUVELLE MACHINE VIRTUELLE WINDOWS

10.15.1. Conditions préalables

- Support d'installation de Windows accessible par la machine virtuelle, par exemple en important une ISO dans un volume de données et en l'attachant à la machine virtuelle.

10.15.2. À propos des pilotes VirtIO

Les pilotes VirtIO sont des pilotes de périphériques paravirtualisés requis pour que les machines virtuelles Microsoft Windows fonctionnent dans OpenShift Virtualization. Les pilotes pris en charge sont disponibles dans le disque de conteneur **container-native-virtualization/virtio-win** du catalogue de l'écosystème Red Hat.

Le disque conteneur **container-native-virtualization/virtio-win** doit être attaché à la machine virtuelle en tant que lecteur CD SATA pour permettre l'installation du pilote. Vous pouvez installer les pilotes VirtIO pendant l'installation de Windows sur la machine virtuelle ou ajoutés à une installation Windows existante.

Une fois les pilotes installés, le disque conteneur **container-native-virtualization/virtio-win** peut être retiré de la machine virtuelle.

Voir aussi : Installation du pilote VirtIO sur une machine virtuelle Windows existante .

10.15.3. Pilotes VirtIO pris en charge pour les machines virtuelles Microsoft Windows

Tableau 10.2. Pilotes pris en charge

Nom du conducteur	ID du matériel	Description
viostor	VEN_1AF4&DEV_1001 VEN_1AF4&DEV_1042	Le pilote de bloc. Se présente parfois sous la forme d'un **SCSI Controller** dans le groupe **Other devices**.
viorng	VEN_1AF4&DEV_1005 VEN_1AF4&DEV_1044	Le pilote de la source d'entropie. S'affiche parfois sous la forme d'un **PCI Device** dans le groupe **Other devices**.
NetKVM	VEN_1AF4&DEV_1000 VEN_1AF4&DEV_1041	Le pilote de réseau. S'affiche parfois comme **Ethernet Controller** dans le groupe **Other devices**. Disponible uniquement si un NIC VirtIO est configuré.

10.15.4. Ajout d'un disque conteneur de pilotes VirtIO à une machine virtuelle

OpenShift Virtualization distribue les pilotes VirtIO pour Microsoft Windows sous la forme d'un disque conteneur, qui est disponible à partir du catalogue de l'écosystème Red Hat . Pour installer ces pilotes sur une machine virtuelle Windows, attachez le disque conteneur **container-native-virtualization/virtio-win** à la machine virtuelle en tant que lecteur de CD SATA dans le fichier de configuration de la machine virtuelle.

Conditions préalables

- Téléchargez le disque de conteneur **container-native-virtualization/virtio-win** à partir du catalogue de l'écosystème Red Hat . Ceci n'est pas obligatoire, car le disque conteneur sera téléchargé depuis le registre Red Hat s'il n'est pas déjà présent dans le cluster, mais cela peut réduire le temps d'installation.

Procédure

1. Ajoutez le disque conteneur **container-native-virtualization/virtio-win** en tant que disque **cdrom** dans le fichier de configuration de la machine virtuelle Windows. Le disque conteneur sera téléchargé depuis le registre s'il n'est pas déjà présent dans le cluster.

```
spec:
  domain:
    devices:
      disks:
        - name: virtiocontainerdisk
          bootOrder: 2 (1)
          cdrom:
            bus: sata
volumes:
  - containerDisk:
      image: container-native-virtualization/virtio-win
    name: virtiocontainerdisk
```

1. OpenShift Virtualization démarre les disques de la machine virtuelle dans l'ordre défini dans le fichier de configuration **VirtualMachine**. Vous pouvez soit définir d'autres disques pour la machine virtuelle avant le disque du conteneur **container-native-virtualization/virtio-win**, soit utiliser le paramètre facultatif **bootOrder** pour vous assurer que la machine virtuelle démarre à partir du bon disque. Si vous spécifiez le paramètre **bootOrder** pour un disque, il doit être spécifié pour tous les disques de la configuration.

2. Le disque est disponible dès que la machine virtuelle a démarré :

   - Si vous ajoutez le disque conteneur à une machine virtuelle en cours d'exécution, utilisez **oc apply -f <vm.yaml>** dans l'interface CLI ou redémarrez la machine virtuelle pour que les modifications soient prises en compte.

   - Si la machine virtuelle n'est pas en cours d'exécution, utilisez **virtctl start <vm>**.

Après le démarrage de la machine virtuelle, les pilotes VirtIO peuvent être installés à partir du lecteur CD SATA connecté.

10.15.5. Installation des pilotes VirtIO pendant l'installation de Windows

Installer les pilotes VirtIO à partir du pilote du CD SATA pendant l'installation de Windows.

NOTE

Cette procédure utilise une approche générique de l'installation de Windows et la méthode d'installation peut différer d'une version à l'autre de Windows. Consultez la documentation de la version de Windows que vous installez.

Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.
2. Commencez le processus d'installation de Windows.
3. Sélectionnez l'installation **Advanced**.
4. La destination de stockage ne sera pas reconnue tant que le pilote n'aura pas été chargé. Cliquez sur **Load driver**.
5. Les pilotes sont attachés à un lecteur de CD SATA. Cliquez sur **OK** et recherchez dans le lecteur de CD le pilote de stockage à charger. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.
6. Répétez les deux étapes précédentes pour tous les pilotes nécessaires.
7. Terminer l'installation de Windows.

10.15.6. Suppression du disque du conteneur VirtIO d'une machine virtuelle

Après avoir installé tous les pilotes VirtIO requis sur la machine virtuelle, le disque conteneur **container-native-virtualization/virtio-win** n'a plus besoin d'être attaché à la machine virtuelle. Supprimez le disque conteneur **container-native-virtualization/virtio-win** du fichier de configuration de la machine virtuelle.

Procédure

1. Modifiez le fichier de configuration et supprimez **disk** et **volume**.

```
$ oc edit vm <vm-name>
```

```
spec:
  domain:
    devices:
      disks:
        - name: virtiocontainerdisk
          bootOrder: 2
          cdrom:
            bus: sata
volumes:
  - containerDisk:
      image: container-native-virtualization/virtio-win
    name: virtiocontainerdisk
```

2. Redémarrez la machine virtuelle pour que les modifications soient prises en compte.

10.16. UTILISATION DE DISPOSITIFS VIRTUELS TRUSTED PLATFORM MODULE

Ajoutez un dispositif Trusted Platform Module (vTPM) virtuel à une machine virtuelle nouvelle ou existante en modifiant le manifeste **VirtualMachine** (VM) ou **VirtualMachineInstance** (VMI).

10.16.1. À propos des dispositifs vTPM

Un dispositif virtuel Trusted Platform Module (vTPM) fonctionne comme une puce matérielle physique Trusted Platform Module (TPM).

Vous pouvez utiliser un périphérique vTPM avec n'importe quel système d'exploitation, mais Windows 11 nécessite la présence d'une puce TPM pour s'installer ou démarrer. Un périphérique vTPM permet aux machines virtuelles créées à partir d'une image Windows 11 de fonctionner sans puce TPM physique.

Si vous n'activez pas vTPM, la VM ne reconnaît pas de périphérique TPM, même si le nœud en possède un.

les dispositifs vTPM protègent également les machines virtuelles en stockant temporairement les secrets sans matériel physique. Cependant, l'utilisation de vTPM pour le stockage de secrets persistants n'est pas prise en charge actuellement. vTPM se débarrasse des secrets stockés après l'arrêt d'une machine virtuelle.

10.16.2. Ajout d'un périphérique vTPM à une machine virtuelle

L'ajout d'un dispositif Trusted Platform Module (vTPM) virtuel à une machine virtuelle (VM) permet d'exécuter une VM créée à partir d'une image Windows 11 sans dispositif TPM physique. Un dispositif vTPM stocke également temporairement les secrets de cette VM.

Procédure

1. Exécutez la commande suivante pour mettre à jour la configuration de la VM :

```
$ oc edit vm <vm_name>
```

2. Modifiez la VM **spec** de manière à inclure la ligne **tpm: {}**. Par exemple :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
   name: example-vm
spec:
 template:
   spec:
     domain:
       devices:
         tpm: {} 1
...
```

1 Ajoute le dispositif TPM à la VM.

3. Pour appliquer vos modifications, enregistrez et quittez l'éditeur.

4. Facultatif : si vous modifiez une machine virtuelle en cours d'exécution, vous devez la redémarrer pour que les modifications soient prises en compte.

10.17. GÉRER LES MACHINES VIRTUELLES AVEC OPENSHIFT PIPELINES

Red Hat OpenShift Pipelines est un framework CI/CD natif Kubernetes qui permet aux développeurs de concevoir et d'exécuter chaque étape du pipeline CI/CD dans son propre conteneur.

L'opérateur de tâches Tekton (TTO) intègre la virtualisation OpenShift avec les pipelines OpenShift. TTO inclut des tâches de cluster et des exemples de pipelines qui vous permettent de :

- Créer et gérer des machines virtuelles (VM), des réclamations de volumes persistants (PVC) et des volumes de données
- Exécuter des commandes dans des machines virtuelles
- Manipuler les images de disque avec les outils **libguestfs**

IMPORTANT

La gestion des machines virtuelles avec Red Hat OpenShift Pipelines est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir des commentaires pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

10.17.1. Conditions préalables

- Vous avez accès à un cluster OpenShift Container Platform avec les permissions **cluster-admin**.
- Vous avez installé l'OpenShift CLI (**oc**).
- Vous avez installé OpenShift Pipelines.

10.17.2. Déployer les ressources de l'opérateur Tekton Tasks

Les tâches du cluster Tekton Tasks Operator (TTO) et les exemples de pipelines ne sont pas déployés par défaut lorsque vous installez OpenShift Virtualization. Pour déployer les ressources TTO, activez la porte de fonctionnalité **deployTektonTaskResources** dans la ressource personnalisée (CR) **HyperConverged**.

Procédure

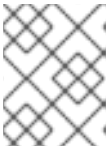
1. Ouvrez le CR **HyperConverged** dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Définissez le champ **spec.featureGates.deployTektonTaskResources** sur **true**.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: kubevirt-hyperconverged
spec:
  tektonPipelinesNamespace: <user_namespace> 1
  featureGates:
    deployTektonTaskResources: true 2
#...
```

1. L'espace de noms dans lequel les pipelines doivent être exécutés.
2. Le portail de fonctionnalités doit être activé pour déployer des ressources TTO.



NOTE

Les tâches de cluster et les exemples de pipelines restent disponibles même si vous désactivez le portail de fonctionnalités ultérieurement.

3. Enregistrez vos modifications et quittez l'éditeur.

10.17.3. Tâches de la machine virtuelle prises en charge par l'opérateur de tâches Tekton

Le tableau suivant présente les tâches de la grappe qui sont incluses dans l'opérateur de tâches Tekton.

Tableau 10.3. Tâches de la machine virtuelle prises en charge par l'opérateur de tâches Tekton

Tâche	Description
create-vm-from-template	Créer une machine virtuelle à partir d'un modèle.
copy-template	Copier un modèle de machine virtuelle.
modify-vm-template	Modifier un modèle de machine virtuelle.
modify-data-object	Créer ou supprimer des volumes de données ou des sources de données.
cleanup-vm	Exécuter un script ou une commande dans une machine virtuelle et arrêter ou supprimer la machine virtuelle par la suite.
disk-virt-customize	Utilisez l'outil **virt-customize** pour exécuter un script de personnalisation sur un PVC cible.
disk-virt-sysprep	Utilisez l'outil **virt-sysprep** pour exécuter un script sysprep sur un PVC cible.
wait-for-vmi-status	Attendre un état spécifique d'une instance de machine virtuelle et échouer ou réussir en fonction de l'état.

10.17.4. Exemples de pipelines

Tekton Tasks Operator comprend les exemples suivants de manifestes **Pipeline**. Vous pouvez exécuter les exemples de pipelines en utilisant la console web ou le CLI.

Pipeline d'installation de Windows 10

Ce pipeline installe Windows 10 dans un nouveau volume de données à partir d'une image d'installation Windows (fichier ISO). Un fichier de réponse personnalisé est utilisé pour exécuter le processus d'installation.

Personnaliser le pipeline de Windows 10

Ce pipeline clone le volume de données d'une installation de base de Windows 10, le personnalise en installant Microsoft SQL Server Express, puis crée une nouvelle image et un nouveau modèle.

10.17.4.1. Exécuter les pipelines d'exemple à l'aide de la console web

Vous pouvez exécuter les exemples de pipelines à partir du menu **Pipelines** de la console web.

Procédure

1. Cliquez sur **Pipelines → Pipelines** dans le menu latéral.
2. Sélectionnez une canalisation pour ouvrir la page **Pipeline details**.
3. Dans la liste **Actions**, sélectionnez **Start**. La boîte de dialogue **Start Pipeline** s'affiche.

4. Conservez les valeurs par défaut des paramètres, puis cliquez sur **Start** pour lancer le pipeline. L'onglet **Details** suit la progression de chaque tâche et affiche l'état du pipeline.

10.17.4.2. Exécuter les pipelines d'exemple à l'aide de l'interface de programmation

Utilisez une ressource **PipelineRun** pour exécuter les exemples de pipelines. Un objet **PipelineRun** est l'instance en cours d'exécution d'un pipeline. Il instancie un pipeline pour l'exécuter avec des entrées, des sorties et des paramètres d'exécution spécifiques sur un cluster. Il crée également un objet **TaskRun** pour chaque tâche du pipeline.

Procédure

1. Pour exécuter le pipeline d'installation de Windows 10, créez le manifeste **PipelineRun** suivant :

```
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
metadata:
  generateName: windows10-installer-run-
  labels:
    pipelinerun: windows10-installer-run
spec:
  params:
  - name: winImageDownloadURL
    value: <link_to_windows_10_iso> (1)
  pipelineRef:
    name: windows10-installer
  taskRunSpecs:
    - pipelineTaskName: copy-template
      taskServiceAccountName: copy-template-task
    - pipelineTaskName: modify-vm-template
      taskServiceAccountName: modify-vm-template-task
    - pipelineTaskName: create-vm-from-template
      taskServiceAccountName: create-vm-from-template-task
    - pipelineTaskName: wait-for-vmi-status
      taskServiceAccountName: wait-for-vmi-status-task
    - pipelineTaskName: create-base-dv
      taskServiceAccountName: modify-data-object-task
    - pipelineTaskName: cleanup-vm
      taskServiceAccountName: cleanup-vm-task
  status: {}
```

(1) Indiquez l'URL du fichier ISO de Windows 10 64 bits. La langue du produit doit être l'anglais (États-Unis).

2. Appliquer le manifeste **PipelineRun**:

```
$ oc apply -f windows10-installer-run.yaml
```

3. Pour exécuter le pipeline de personnalisation de Windows 10, créez le manifeste **PipelineRun** suivant :

```
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
metadata:
```

```
    generateName: windows10-customize-run-
    labels:
      pipelinerun: windows10-customize-run
  spec:
    params:
      - name: allowReplaceGoldenTemplate
        value: true
      - name: allowReplaceCustomizationTemplate
        value: true
    pipelineRef:
      name: windows10-customize
    taskRunSpecs:
      - pipelineTaskName: copy-template-customize
        taskServiceAccountName: copy-template-task
      - pipelineTaskName: modify-vm-template-customize
        taskServiceAccountName: modify-vm-template-task
      - pipelineTaskName: create-vm-from-template
        taskServiceAccountName: create-vm-from-template-task
      - pipelineTaskName: wait-for-vmi-status
        taskServiceAccountName: wait-for-vmi-status-task
      - pipelineTaskName: create-base-dv
        taskServiceAccountName: modify-data-object-task
      - pipelineTaskName: cleanup-vm
        taskServiceAccountName: cleanup-vm-task
      - pipelineTaskName: copy-template-golden
        taskServiceAccountName: copy-template-task
      - pipelineTaskName: modify-vm-template-golden
        taskServiceAccountName: modify-vm-template-task
  status: {}
```

4. Appliquer le manifeste **PipelineRun**:

```
$ oc apply -f windows10-customize-run.yaml
```

10.17.5. Ressources supplémentaires

- Création de solutions CI/CD pour les applications à l'aide de Red Hat OpenShift Pipelines

10.18. GESTION AVANCÉE DES MACHINES VIRTUELLES

10.18.1. Travailler avec des quotas de ressources pour les machines virtuelles

Créer et gérer des quotas de ressources pour les machines virtuelles.

10.18.1.1. Définir des limites de quotas de ressources pour les machines virtuelles

Les quotas de ressources qui n'utilisent que des demandes fonctionnent automatiquement avec les machines virtuelles (VM). Si votre quota de ressources utilise des limites, vous devez définir manuellement des limites de ressources sur les machines virtuelles. Les limites de ressources doivent être supérieures d'au moins 100 Mo aux demandes de ressources.

Procédure

1. Définissez des limites pour une VM en modifiant le manifeste **VirtualMachine**. Par exemple :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: with-limits
spec:
  running: false
  template:
    spec:
      domain:
# ...
        resources:
          requests:
            memory: 128Mi
          limits:
            memory: 256Mi 1
```

1 Cette configuration est supportée parce que la valeur **limits.memory** est au moins **100Mi** plus grande que la valeur **requests.memory**.

2. Sauvegarder le manifeste **VirtualMachine**.

10.18.1.2. Ressources supplémentaires

- Quotas de ressources par projet
- Quotas de ressources pour plusieurs projets

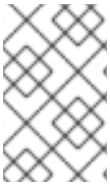
10.18.2. Spécifier des nœuds pour les machines virtuelles

Vous pouvez placer des machines virtuelles (VM) sur des nœuds spécifiques en utilisant des règles de placement de nœuds.

10.18.2.1. À propos du placement des nœuds pour les machines virtuelles

Pour garantir que les machines virtuelles (VM) s'exécutent sur les nœuds appropriés, vous pouvez configurer des règles de placement des nœuds. Cette opération peut s'avérer utile dans les cas suivants

- Vous avez plusieurs machines virtuelles. Pour garantir la tolérance aux pannes, vous souhaitez qu'elles s'exécutent sur des nœuds différents.
- Vous avez deux machines virtuelles bavardes. Pour éviter la redondance du routage inter-nœuds, vous souhaitez que les VM s'exécutent sur le même nœud.
- Vos machines virtuelles nécessitent des caractéristiques matérielles spécifiques qui ne sont pas présentes sur tous les nœuds disponibles.
- Vous avez un module qui ajoute des capacités à un nœud et vous voulez placer une VM sur ce nœud pour qu'elle puisse utiliser ces capacités.



NOTE

Le placement des machines virtuelles s'appuie sur les règles de placement des charges de travail dans les nœuds. Si des charges de travail sont exclues de certains nœuds au niveau du composant, les machines virtuelles ne peuvent pas être placées sur ces nœuds.

Vous pouvez utiliser les types de règles suivants dans le champ **spec** d'un manifeste **VirtualMachine**:

nodeSelector

Permet de planifier des machines virtuelles sur des nœuds étiquetés avec la ou les paires clé-valeur spécifiées dans ce champ. Les étiquettes du nœud doivent correspondre exactement à toutes les paires répertoriées.

affinity

Permet d'utiliser une syntaxe plus expressive pour définir des règles qui font correspondre les nœuds aux machines virtuelles. Par exemple, vous pouvez spécifier qu'une règle est une préférence plutôt qu'une exigence absolue, de sorte que les machines virtuelles soient toujours planifiées si la règle n'est pas respectée. L'affinité de pod, l'anti-affinité de pod et l'affinité de nœud sont prises en charge pour le placement des machines virtuelles. L'affinité de pod fonctionne pour les machines virtuelles car le type de charge de travail **VirtualMachine** est basé sur l'objet **Pod**.

NOTE

Les règles d'affinité ne s'appliquent que lors de la planification. OpenShift Container Platform ne replanifie pas les charges de travail en cours d'exécution si les contraintes ne sont plus respectées.

tolerations

Permet aux machines virtuelles d'être planifiées sur des nœuds qui ont des taches correspondantes. Si une erreur est appliquée à un nœud, ce nœud n'accepte que les machines virtuelles qui tolèrent l'erreur.

10.18.2.2. Exemples de placement de nœuds

Les exemples suivants d'extraits de fichiers YAML utilisent les champs **nodePlacement**, **affinity** et **tolerations** pour personnaliser l'emplacement des nœuds pour les machines virtuelles.

10.18.2.2.1. Exemple : Placement de nœuds de VM avec nodeSelector

Dans cet exemple, la machine virtuelle a besoin d'un nœud dont les métadonnées contiennent des étiquettes **example-key-1 = example-value-1** et **example-key-2 = example-value-2**.

AVERTISSEMENT

Si aucun nœud ne correspond à cette description, la machine virtuelle n'est pas planifiée.

Exemple de manifeste VM

```
-
```

```
metadata:
  name: example-vm-node-selector
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  template:
    spec:
      nodeSelector:
        example-key-1: example-value-1
        example-key-2: example-value-2
...
```

10.18.2.2.2. Exemple : Placement de nœuds VM avec affinité de pod et anti-affinité de pod

Dans cet exemple, la VM doit être programmée sur un nœud disposant d'un pod en cours d'exécution portant l'étiquette **example-key-1 = example-value-1**. Si aucun pod de ce type n'est en cours d'exécution sur un nœud, la VM n'est pas programmée.

Dans la mesure du possible, la VM n'est pas programmée sur un nœud qui possède un pod portant l'étiquette **example-key-2 = example-value-2**. Cependant, si tous les nœuds candidats ont un pod avec ce label, l'ordonnanceur ignore cette contrainte.

Exemple de manifeste VM

```
metadata:
  name: example-vm-pod-affinity
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution: 1
      - labelSelector:
          matchExpressions:
          - key: example-key-1
            operator: In
            values:
            - example-value-1
        topologyKey: kubernetes.io/hostname
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution: 2
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: example-key-2
              operator: In
              values:
              - example-value-2
          topologyKey: kubernetes.io/hostname
...
```

1. Si vous utilisez le type de règle **requiredDuringSchedulingIgnoredDuringExecution**, la VM n'est pas planifiée si la contrainte n'est pas respectée.

2 Si vous utilisez le type de règle **preferredDuringSchedulingIgnoredDuringExecution**, la VM est toujours planifiée si la contrainte n'est pas respectée, à condition que toutes les contraintes

10.18.2.2.3. Exemple : Placement de nœuds de VM avec affinité de nœuds

Dans cet exemple, la VM doit être programmée sur un nœud portant l'étiquette **example.io/example-key = example-value-1** ou l'étiquette **example.io/example-key = example-value-2**. La contrainte est respectée si une seule des étiquettes est présente sur le nœud. Si aucune étiquette n'est présente, la VM n'est pas programmée.

Dans la mesure du possible, l'ordonnanceur évite les nœuds portant l'étiquette **example-node-label-key = example-node-label-value**. Toutefois, si tous les nœuds candidats portent cette étiquette, l'ordonnanceur ignore cette contrainte.

Exemple de manifeste VM

```
metadata:
  name: example-vm-node-affinity
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution: 1
        nodeSelectorTerms:
        - matchExpressions:
          - key: example.io/example-key
            operator: In
            values:
            - example-value-1
            - example-value-2
      preferredDuringSchedulingIgnoredDuringExecution: 2
      - weight: 1
        preference:
          matchExpressions:
          - key: example-node-label-key
            operator: In
            values:
            - example-node-label-value
...
```

1 Si vous utilisez le type de règle **requiredDuringSchedulingIgnoredDuringExecution**, la VM n'est pas planifiée si la contrainte n'est pas respectée.

2 Si vous utilisez le type de règle **preferredDuringSchedulingIgnoredDuringExecution**, la VM est toujours planifiée si la contrainte n'est pas respectée, à condition que toutes les contraintes requises soient respectées.

10.18.2.2.4. Exemple : Placement de nœuds de VM avec tolérances

Dans cet exemple, les nœuds réservés aux machines virtuelles sont déjà marqués de l'erreur **key=virtualization:NoSchedule**. Étant donné que cette machine virtuelle a une adresse **tolerations** correspondante, elle peut planifier sur les nœuds entachés.

NOTE

Une machine virtuelle qui tolère une erreur n'est pas obligée de planifier sur un nœud avec cette erreur.

Exemple de manifeste VM

```
metadata:
  name: example-vm-tolerations
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "virtualization"
    effect: "NoSchedule"
...
```

10.18.2.3. Ressources supplémentaires

- Spécification des nœuds pour les composants de virtualisation
- Placer des pods sur des nœuds spécifiques en utilisant des sélecteurs de nœuds
- Contrôle du placement des pods sur les nœuds à l'aide de règles d'affinité des nœuds
- Contrôle du placement de pods à l'aide de taches de nœuds

10.18.3. Configuration de la rotation des certificats

Configurer les paramètres de rotation des certificats pour remplacer les certificats existants.

10.18.3.1. Configuration de la rotation des certificats

Vous pouvez le faire pendant l'installation d'OpenShift Virtualization dans la console web ou après l'installation dans la ressource personnalisée (CR) **HyperConverged**.

Procédure

1. Ouvrez le CR **HyperConverged** en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Modifiez les champs **spec.certConfig** comme indiqué dans l'exemple suivant. Pour éviter de surcharger le système, assurez-vous que toutes les valeurs sont supérieures ou égales à 10 minutes. Exprimez toutes les valeurs sous forme de chaînes de caractères conformes au format golang **ParseDuration** .

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
 name: kubevirt-hyperconverged
 namespace: openshift-cnv
```

```
spec:
  certConfig:
    ca:
      duration: 48h0m0s
      renewBefore: 24h0m0s (1)
    server:
      duration: 24h0m0s (2)
      renewBefore: 12h0m0s (3)
```

1. La valeur de **ca.renewBefore** doit être inférieure ou égale à la valeur de **ca.duration**.
2. La valeur de **server.duration** doit être inférieure ou égale à la valeur de **ca.duration**.
3. La valeur de **server.renewBefore** doit être inférieure ou égale à la valeur de **server.duration**.

3. Appliquez le fichier YAML à votre cluster.

10.18.3.2. Dépannage des paramètres de rotation des certificats

La suppression d'une ou de plusieurs valeurs de **certConfig** entraîne le retour aux valeurs par défaut, à moins que les valeurs par défaut n'entrent en conflit avec l'une des conditions suivantes :

- La valeur de **ca.renewBefore** doit être inférieure ou égale à la valeur de **ca.duration**.
- La valeur de **server.duration** doit être inférieure ou égale à la valeur de **ca.duration**.
- La valeur de **server.renewBefore** doit être inférieure ou égale à la valeur de **server.duration**.

Si les valeurs par défaut sont en conflit avec ces conditions, vous recevrez un message d'erreur.

Si vous supprimez la valeur **server.duration** dans l'exemple suivant, la valeur par défaut de **24h0m0s** est supérieure à la valeur de **ca.duration**, ce qui est contraire aux conditions spécifiées.

Exemple :

```
certConfig:
  ca:
    duration: 4h0m0s
    renewBefore: 1h0m0s
  server:
    duration: 4h0m0s
    renewBefore: 4h0m0s
```

Il en résulte le message d'erreur suivant :

```
error: hyperconvergeds.hco.kubevirt.io "kubevirt-hyperconverged" could not be patched: admission webhook "validate-hco.kubevirt.io" denied the request: spec.certConfig: ca.duration is smaller than server.duration
```

Le message d'erreur ne mentionne que le premier conflit. Vérifiez toutes les valeurs de certConfig avant de poursuivre.

10.18.4. Utilisation du mode UEFI pour les machines virtuelles

Vous pouvez démarrer une machine virtuelle (VM) en mode UEFI (Unified Extensible Firmware Interface).

10.18.4.1. À propos du mode UEFI pour les machines virtuelles

L'Unified Extensible Firmware Interface (UEFI), à l'instar de l'ancien BIOS, initialise les composants matériels et les fichiers image du système d'exploitation au démarrage de l'ordinateur. L'UEFI prend en charge des fonctions et des options de personnalisation plus modernes que le BIOS, ce qui permet des temps de démarrage plus rapides.

Il stocke toutes les informations relatives à l'initialisation et au démarrage dans un fichier portant l'extension **.efi**, qui est stocké sur une partition spéciale appelée partition système EFI (ESP). L'ESP contient également les programmes de démarrage du système d'exploitation installé sur l'ordinateur.

10.18.4.2. Démarrage des machines virtuelles en mode UEFI

Vous pouvez configurer une machine virtuelle pour qu'elle démarre en mode UEFI en modifiant le manifeste **VirtualMachine**.

Conditions préalables

- Installez le CLI OpenShift (**oc**).

Procédure

1. Modifiez ou créez un fichier manifeste **VirtualMachine**. Utilisez la strophe **spec.firmware.bootloader** pour configurer le mode UEFI :

   Démarrage en mode UEFI avec démarrage sécurisé actif

```
apiversion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    special: vm-secureboot
  name: vm-secureboot
spec:
  template:
    metadata:
      labels:
        special: vm-secureboot
    spec:
      domain:
        devices:
          disks:
          - disk:
              bus: virtio
            name: containerdisk
        features:
          acpi: {}
          smm:
            enabled: true (1)
        firmware:
```

```
        bootloader:
          efi:
            secureBoot: true 2
...
```

1. OpenShift Virtualization nécessite que le mode de gestion du système (**SMM**) soit activé pour que le démarrage sécurisé en mode UEFI se produise.
2. OpenShift Virtualization supporte une VM avec ou sans Secure Boot lors de l'utilisation du mode UEFI. Si Secure Boot est activé, le mode UEFI est requis. Cependant, le mode UEFI peut être activé sans utiliser le mode Secure Boot.

2. Appliquez le manifeste à votre cluster en exécutant la commande suivante :

```
oc create -f <nom_du_fichier>.yaml
```

10.18.5. Configuration du démarrage PXE pour les machines virtuelles

Le démarrage PXE, ou démarrage en réseau, est disponible dans OpenShift Virtualization. Le démarrage en réseau permet à un ordinateur de démarrer et de charger un système d'exploitation ou un autre programme sans nécessiter de périphérique de stockage connecté localement. Par exemple, vous pouvez l'utiliser pour choisir votre image de système d'exploitation souhaitée à partir d'un serveur PXE lors du déploiement d'un nouvel hôte.

10.18.5.1. Conditions préalables

- Un pont Linux doit être connecté.
- Le serveur PXE doit être connecté au même VLAN que le pont.

10.18.5.2. Démarrage PXE avec une adresse MAC spécifiée

En tant qu'administrateur, vous pouvez démarrer un client sur le réseau en créant d'abord un objet **NetworkAttachmentDefinition** pour votre réseau PXE. Ensuite, faites référence à la définition de l'attachement réseau dans votre fichier de configuration de l'instance de machine virtuelle avant de démarrer l'instance de machine virtuelle. Vous pouvez également spécifier une adresse MAC dans le fichier de configuration de l'instance de machine virtuelle, si le serveur PXE l'exige.

Conditions préalables

- Un pont Linux doit être connecté.
- Le serveur PXE doit être connecté au même VLAN que le pont.

Procédure

1. Configurer un réseau PXE sur le cluster :
   a. Créer le fichier de définition de l'attachement réseau pour le réseau PXE **pxe-net-conf**:

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
```

```
  name: pxe-net-conf
spec:
  config: '{
    "cniVersion": "0.3.1",
    "name": "pxe-net-conf",
    "plugins": [
      {
        "type": "cnv-bridge",
        "bridge": "br1",
        "vlan": 1 (1)
      },
      {
        "type": "cnv-tuning" (2)
      }
    ]
  }'
```

(1) Facultatif : La balise VLAN.

(2) Le plugin **cnv-tuning** prend en charge les adresses MAC personnalisées.

NOTE

L'instance de machine virtuelle sera connectée au pont **br1** via un port d'accès avec le VLAN demandé.

2. Créez la définition de la pièce jointe au réseau en utilisant le fichier que vous avez créé à l'étape précédente :

```
$ oc create -f pxe-net-conf.yaml
```

3. Modifiez le fichier de configuration de l'instance de la machine virtuelle pour y inclure les détails de l'interface et du réseau.

   a. Spécifiez le réseau et l'adresse MAC, si le serveur PXE l'exige. Si l'adresse MAC n'est pas spécifiée, une valeur est attribuée automatiquement.
   Assurez-vous que **bootOrder** est défini sur **1** afin que l'interface démarre en premier. Dans cet exemple, l'interface est connectée à un réseau appelé **<pxe-net>**:

```
interfaces:
- masquerade: {}
  name: default
- bridge: {}
  name: pxe-net
  macAddress: de:00:00:00:00:de
  bootOrder: 1
```

NOTE

L'ordre de démarrage est global pour les interfaces et les disques.

b. Attribuez un numéro de périphérique de démarrage au disque afin de garantir un démarrage correct après le provisionnement du système d'exploitation.
Réglez la valeur du disque **bootOrder** sur **2**:

```
devices:
  disks:
  - disk:
      bus: virtio
    name: containerdisk
    bootOrder: 2
```

c. Spécifiez que le réseau est connecté à la définition de l'attachement au réseau créée précédemment. Dans ce scénario, **<pxe-net>** est connecté à la définition d'attachement au réseau appelée **<pxe-net-conf>**:

```
networks:
- name: default
  pod: {}
- name: pxe-net
  multus:
    networkName: pxe-net-conf
```

4. Créer l'instance de la machine virtuelle :

```
$ oc create -f vmi-pxe-boot.yaml
```

Exemple de sortie

```
virtualmachineinstance.kubevirt.io "vmi-pxe-boot" created
```

1. Attendez que l'instance de la machine virtuelle s'exécute :

```
$ oc get vmi vmi-pxe-boot -o yaml | grep -i phase
  phase: Running
```

2. Visualisez l'instance de la machine virtuelle à l'aide de VNC :

```
$ virtctl vnc vmi-pxe-boot
```

3. Observez l'écran de démarrage pour vérifier que le démarrage PXE est réussi.

4. Connectez-vous à l'instance de la machine virtuelle :

```
$ virtctl console vmi-pxe-boot
```

5. Vérifiez les interfaces et l'adresse MAC sur la machine virtuelle et que l'interface connectée au pont a l'adresse MAC spécifiée. Dans ce cas, nous avons utilisé **eth1** pour le démarrage PXE, sans adresse IP. L'autre interface, **eth0**, a reçu une adresse IP de OpenShift Container Platform.

```
$ ip addr
```

Exemple de sortie

```
-
```

```
...
3. eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether de:00:00:00:00:de brd ff:ff:ff:ff:ff:ff
```

10.18.5.3. Glossaire de la mise en réseau de la virtualisation OpenShift

OpenShift Virtualization offre des fonctionnalités avancées de mise en réseau en utilisant des ressources et des plugins personnalisés.

Les termes suivants sont utilisés dans la documentation d'OpenShift Virtualization :

Interface de réseau de conteneurs (CNI)

un projet de la Cloud Native Computing Foundation, axé sur la connectivité réseau des conteneurs. OpenShift Virtualization utilise des plugins CNI pour s'appuyer sur la fonctionnalité réseau de base de Kubernetes.

Multus

un plugin CNI "meta" Plugin CNI qui permet à plusieurs CNI d'exister afin qu'un pod ou une machine virtuelle puisse utiliser les interfaces dont il a besoin.

Définition des ressources personnalisées (CRD)

une ressource de l'API Kubernetes qui vous permet de définir des ressources personnalisées, ou un objet défini à l'aide de la ressource de l'API CRD.

Définition de l'attachement au réseau (NAD)

un CRD introduit par le projet Multus qui permet d'attacher des pods, des machines virtuelles et des instances de machines virtuelles à un ou plusieurs réseaux.

Politique de configuration du réseau de nœuds (NNCP)

une description de la configuration réseau requise sur les nœuds. Vous mettez à jour la configuration du réseau des nœuds, notamment en ajoutant ou en supprimant des interfaces, en appliquant un manifeste **NodeNetworkConfigurationPolicy** à la grappe.

Environnement d'exécution avant démarrage (PXE)

une interface qui permet à un administrateur de démarrer une machine cliente à partir d'un serveur via le réseau. Le démarrage en réseau permet de charger à distance des systèmes d'exploitation et d'autres logiciels sur le client.

10.18.6. Utilisation de pages volumineuses avec des machines virtuelles

Vous pouvez utiliser d'énormes pages comme mémoire de secours pour les machines virtuelles de votre cluster.

10.18.6.1. Conditions préalables

- Les nœuds doivent avoir des pages énormes pré-allouées configurées.

10.18.6.2. Ce que font les grandes pages

La mémoire est gérée par blocs appelés pages. Sur la plupart des systèmes, une page correspond à 4Ki. 1Mi de mémoire équivaut à 256 pages ; 1Gi de mémoire équivaut à 256 000 pages, et ainsi de suite. Les unités centrales de traitement disposent d'une unité de gestion de la mémoire intégrée qui gère une liste de ces pages au niveau matériel. Le Translation Lookaside Buffer (TLB) est une petite mémoire cache matérielle des correspondances entre les pages virtuelles et les pages physiques. Si l'adresse virtuelle transmise dans une instruction matérielle peut être trouvée dans le TLB, la correspondance peut être déterminée rapidement. Si ce n'est pas le cas, la TLB est manquée et le système revient à une

traduction d'adresse plus lente, basée sur le logiciel, ce qui entraîne des problèmes de performance. La taille de la TLB étant fixe, le seul moyen de réduire le risque d'erreur de la TLB est d'augmenter la taille de la page.

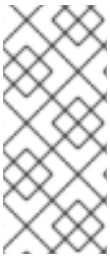
Une page énorme est une page de mémoire dont la taille est supérieure à 4Ki. Sur les architectures x86_64, il existe deux tailles courantes de pages énormes : 2Mi et 1Gi. Les tailles varient sur les autres architectures. Pour utiliser les pages énormes, le code doit être écrit de manière à ce que les applications en soient conscientes. Les pages énormes transparentes (THP) tentent d'automatiser la gestion des pages énormes sans que l'application en ait connaissance, mais elles ont des limites. En particulier, elles sont limitées à des tailles de page de 2Mi. Les THP peuvent entraîner une dégradation des performances sur les nœuds à forte utilisation ou fragmentation de la mémoire en raison des efforts de défragmentation des THP, qui peuvent bloquer les pages de mémoire. Pour cette raison, certaines applications peuvent être conçues pour (ou recommander) l'utilisation d'énormes pages pré-allouées au lieu de THP.

Dans OpenShift Virtualization, les machines virtuelles peuvent être configurées pour consommer des pages énormes pré-allouées.

10.18.6.3. Configuration de pages volumineuses pour les machines virtuelles

Vous pouvez configurer les machines virtuelles pour qu'elles utilisent des pages énormes pré-allouées en incluant les paramètres **memory.hugepages.pageSize** et **resources.requests.memory** dans la configuration de votre machine virtuelle.

La demande de mémoire doit être divisible par la taille de la page. Par exemple, vous ne pouvez pas demander la mémoire **500Mi** avec une taille de page de **1Gi**.



NOTE

Les configurations de la mémoire de l'hôte et du système d'exploitation invité ne sont pas liées. Les pages volumineuses demandées dans le manifeste de la machine virtuelle s'appliquent à QEMU. Les pages volumineuses à l'intérieur de l'invité ne peuvent être configurées qu'en fonction de la quantité de mémoire disponible de l'instance de la machine virtuelle.

Si vous modifiez une machine virtuelle en cours d'exécution, celle-ci doit être redémarrée pour que les modifications soient prises en compte.

Conditions préalables

- Les nœuds doivent avoir des pages énormes pré-allouées configurées.

Procédure

1. Dans la configuration de votre machine virtuelle, ajoutez les paramètres **resources.requests.memory** et **memory.hugepages.pageSize** au paramètre **spec.domain**. L'extrait de configuration suivant concerne une machine virtuelle qui demande un total de **4Gi** de mémoire avec une taille de page de **1Gi**:

```
kind: VirtualMachine
...
spec:
  domain:
    resources:
      requests:
```

```
        memory: "4Gi" (1)
      memory:
        hugepages:
          pageSize: "1Gi" (2)
...
```

(1) La quantité totale de mémoire demandée pour la machine virtuelle. Cette valeur doit être divisible par la taille de la page.

(2) La taille de chaque grande page. Les valeurs valables pour l'architecture x86_64 sont **1Gi** et **2Mi**. La taille de la page doit être inférieure à la mémoire demandée.

2. Appliquer la configuration de la machine virtuelle :

```
oc apply -f <virtual_machine>.yaml
```

10.18.7. Activation de ressources dédiées pour les machines virtuelles

Pour améliorer les performances, vous pouvez dédier des ressources de nœuds, telles que l'unité centrale, à une machine virtuelle.

10.18.7.1. À propos des ressources dédiées

Lorsque vous activez les ressources dédiées pour votre machine virtuelle, la charge de travail de votre machine virtuelle est planifiée sur des CPU qui ne seront pas utilisés par d'autres processus. En utilisant des ressources dédiées, vous pouvez améliorer les performances de la machine virtuelle et la précision des prévisions de latence.

10.18.7.2. Conditions préalables

- Le gestionnaire de CPU doit être configuré sur le nœud. Vérifiez que le nœud possède le label **cpumanager = true** avant de planifier les charges de travail des machines virtuelles.
- La machine virtuelle doit être mise hors tension.

10.18.7.3. Activation de ressources dédiées pour une machine virtuelle

Vous activez les ressources dédiées pour une machine virtuelle dans l'onglet **Details**. Les machines virtuelles qui ont été créées à partir d'un modèle Red Hat peuvent être configurées avec des ressources dédiées.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Dans l'onglet **Scheduling**, cliquez sur l'icône représentant un crayon à côté de **Dedicated Resources**.
4. Sélectionnez **Schedule this workload with dedicated resources (guaranteed policy)**

5. Cliquez sur **Save**.

10.18.8. Planification des machines virtuelles

Vous pouvez planifier une machine virtuelle (VM) sur un nœud en vous assurant que le modèle de CPU et l'attribut de stratégie de la VM sont compatibles avec les modèles de CPU et les attributs de stratégie pris en charge par le nœud.

10.18.8.1. Attributs de la politique

Vous pouvez planifier une machine virtuelle (VM) en spécifiant un attribut de stratégie et une caractéristique de CPU qui est mise en correspondance pour la compatibilité lorsque la VM est planifiée sur un nœud. Un attribut de stratégie spécifié pour une VM détermine la manière dont cette VM est planifiée sur un nœud.

Attribut de la politique	Description
force	La VM est obligée d'être programmée sur un nœud. Cela est vrai même si l'unité centrale de l'hôte ne prend pas en charge l'unité centrale de la VM.
exiger	Politique par défaut qui s'applique à une VM si celle-ci n'est pas configurée avec un modèle de CPU spécifique et une spécification de fonctionnalité. Si un nœud n'est pas configuré pour prendre en charge la découverte de nœuds de CPU avec cet attribut de stratégie par défaut ou l'un des autres attributs de stratégie, les VM ne sont pas planifiées sur ce nœud. Le CPU de l'hôte doit prendre en charge le CPU de la VM ou l'hyperviseur doit pouvoir émuler le modèle de CPU pris en charge.
facultatif	La VM est ajoutée à un nœud si elle est prise en charge par le processeur de la machine physique de l'hôte.
désactiver	La VM ne peut pas être planifiée avec la découverte du nœud CPU.
interdire	La VM n'est pas planifiée même si la fonction est prise en charge par l'unité centrale hôte et que la découverte du nœud de l'unité centrale est activée.

10.18.8.2. Définition d'un attribut de politique et d'une fonction de l'unité centrale

Vous pouvez définir un attribut de stratégie et une fonction CPU pour chaque machine virtuelle (VM) afin de vous assurer qu'elle est planifiée sur un nœud conformément à la stratégie et à la fonction. La fonction CPU que vous définissez est vérifiée pour s'assurer qu'elle est prise en charge par le processeur hôte ou émulée par l'hyperviseur.

Procédure

- Modifiez la spécification **domain** de votre fichier de configuration de la VM. L'exemple suivant définit la fonctionnalité CPU et la stratégie **require** pour une machine virtuelle (VM) :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: myvm
```

```
spec:
  template:
    spec:
      domain:
        cpu:
          features:
            - name: apic (1)
              policy: require (2)
```

(1) Nom de la fonction CPU pour la VM.

(2) Attribut de politique pour la VM.

10.18.8.3. Planification des machines virtuelles avec le modèle de CPU pris en charge

Vous pouvez configurer un modèle de CPU pour une machine virtuelle (VM) afin de la planifier sur un nœud où son modèle de CPU est pris en charge.

Procédure

- Modifiez la spécification **domain** du fichier de configuration de votre machine virtuelle. L'exemple suivant montre un modèle de CPU spécifique défini pour une VM :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: myvm
spec:
  template:
    spec:
      domain:
        cpu:
          model: Conroe (1)
```

(1) Modèle de CPU pour la VM.

10.18.8.4. Ordonnancement des machines virtuelles avec le modèle d'hôte

Lorsque le modèle de CPU d'une machine virtuelle (VM) est défini sur **host-model**, la VM hérite du modèle de CPU du nœud où elle est planifiée.

Procédure

- Modifiez la spécification **domain** de votre fichier de configuration de la VM. L'exemple suivant montre que **host-model** est spécifié pour la machine virtuelle :

```
apiVersion: kubevirt/v1alpha3
kind: VirtualMachine
metadata:
  name: myvm
spec:
  template:
```

```
spec:
  domain:
    cpu:
      model: host-model 1
```

1 La VM qui hérite du modèle de CPU du nœud où elle est planifiée.

10.18.9. Configuration de PCI passthrough

La fonction Peripheral Component Interconnect (PCI) passthrough vous permet d'accéder à des périphériques matériels et de les gérer à partir d'une machine virtuelle. Lorsque la fonction PCI passthrough est configurée, les périphériques PCI fonctionnent comme s'ils étaient physiquement connectés au système d'exploitation invité.

Les administrateurs de clusters peuvent exposer et gérer les périphériques hôtes qui sont autorisés à être utilisés dans le cluster en utilisant l'interface de ligne de commande (CLI) **oc**.

10.18.9.1. A propos de la préparation d'un périphérique hôte pour le PCI passthrough

Pour préparer un périphérique hôte au passage PCI à l'aide de l'interface de programmation, créez un objet **MachineConfig** et ajoutez des arguments de noyau pour activer l'unité de gestion de la mémoire d'entrée-sortie (IOMMU). Liez le périphérique PCI au pilote Virtual Function I/O (VFIO), puis exposez-le dans le cluster en modifiant le champ **permittedHostDevices** de la ressource personnalisée (CR) **HyperConverged**. La liste **permittedHostDevices** est vide lorsque vous installez l'opérateur de virtualisation OpenShift pour la première fois.

Pour supprimer un périphérique hôte PCI du cluster à l'aide de la CLI, supprimez les informations relatives au périphérique PCI du CR **HyperConverged**.

10.18.9.1.1. Ajout d'arguments au noyau pour activer le pilote IOMMU

Pour activer le pilote IOMMU (Input-Output Memory Management Unit) dans le noyau, créez l'objet **MachineConfig** et ajoutez les arguments du noyau.

Conditions préalables

- Privilège administratif sur un cluster OpenShift Container Platform en fonctionnement.
- Matériel CPU Intel ou AMD.
- La technologie de virtualisation Intel pour les extensions Directed I/O ou AMD IOMMU dans le BIOS (Basic Input/Output System) est activée.

Procédure

1. Créez un objet **MachineConfig** qui identifie l'argument du noyau. L'exemple suivant montre un argument du noyau pour un processeur Intel.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker 1
  name: 100-worker-iommu 2
```

```
spec:
  config:
    ignition:
      version: 3.2.0
  kernelArguments:
      - intel_iommu=on 3
...
```

1. Applique le nouvel argument du noyau uniquement aux nœuds de travail.
2. L'adresse **name** indique le rang de cet argument du noyau (100) parmi les configurations de la machine et son objectif. Si vous avez un processeur AMD, spécifiez l'argument du noyau comme **amd_iommu=on**.
3. Identifie l'argument du noyau comme étant **intel_iommu** pour un processeur Intel.

2. Créer le nouvel objet **MachineConfig**:

```
$ oc create -f 100-worker-kernel-arg-iommu.yaml
```

Vérification

- Vérifiez que le nouvel objet **MachineConfig** a bien été ajouté.

```
$ oc get MachineConfig
```

10.18.9.1.2. Liaison des périphériques PCI au pilote VFIO

Pour lier les périphériques PCI au pilote VFIO (Virtual Function I/O), obtenez les valeurs de **vendor-ID** et **device-ID** de chaque périphérique et créez une liste avec ces valeurs. Ajoutez cette liste à l'objet **MachineConfig**. L'opérateur **MachineConfig** génère l'objet **/etc/modprobe.d/vfio.conf** sur les nœuds dotés de périphériques PCI et lie les périphériques PCI au pilote VFIO.

Conditions préalables

- Vous avez ajouté des arguments au noyau pour activer l'IOMMU pour le processeur.

Procédure

1. Exécutez la commande **lspci** pour obtenir les adresses **vendor-ID** et **device-ID** du périphérique PCI.

```
$ lspci -nnv | grep -i nvidia
```

Exemple de sortie

```
02:01.0 3D controller [0302]: NVIDIA Corporation GV100GL [Tesla V100 PCIe 32GB]
[10de:1eb8] (rev a1)
```

2. Créer un fichier de configuration Butane, **100-worker-vfiopci.bu**, liant le périphérique PCI au pilote VFIO.

NOTE

See "Creating machine configs with Butane" for information about Butane.

Exemple :

```
variant: openshift
version: 4.12.0
metadata:
  name: 100-worker-vfiopci
  labels:
    machineconfiguration.openshift.io/role: worker 1
storage:
  files:
  - path: /etc/modprobe.d/vfio.conf
    mode: 0644
    overwrite: true
    contents:
      inline: |
        options vfio-pci ids=10de:1eb8 2
  - path: /etc/modules-load.d/vfio-pci.conf 3
    mode: 0644
    overwrite: true
    contents:
      inline: vfio-pci
```

1. Applique le nouvel argument du noyau uniquement aux nœuds de travail.
2. Spécifiez la valeur **vendor-ID** (**10de**) et la valeur **device-ID** (**1eb8**) déterminées précédemment pour lier un seul périphérique au pilote VFIO. Vous pouvez ajouter une liste de plusieurs périphériques avec leurs informations sur le fournisseur et le périphérique.
3. Le fichier qui charge le module noyau vfio-pci sur les nœuds de travail.

3. Utilisez Butane pour générer un fichier objet **MachineConfig**, **100-worker-vfiopci.yaml**, contenant la configuration à fournir aux nœuds de travail :

```
$ butane 100-worker-vfiopci.bu -o 100-worker-vfiopci.yaml
```

4. Appliquer l'objet **MachineConfig** aux nœuds de travail :

```
$ oc apply -f 100-worker-vfiopci.yaml
```

5. Vérifiez que l'objet **MachineConfig** a bien été ajouté.

```
$ oc get MachineConfig
```

Exemple de sortie

```
NAME                GENERATEDBYCONTROLLER                      IGNITIONVERSION
AGE
00-master           d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0          25h
```

```
00-worker                d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0             25h
01-master-container-runtime      d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0
25h
01-master-kubelet                d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0
25h
01-worker-container-runtime      d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0
25h
01-worker-kubelet                d3da910bfa9f4b599af4ed7f5ac270d55950a3a1   3.2.0
25h
100-worker-iommu                                                           3.2.0             30s
100-worker-vfiopci-configuration                                           3.2.0             30s
```

Vérification

- Vérifiez que le pilote VFIO est chargé.

```
$ lspci -nnk -d 10de:
```

La sortie confirme que le pilote VFIO est utilisé.

Exemple de sortie

```
04:00.0 3D controller [0302]: NVIDIA Corporation GP102GL [Tesla P40] [10de:1eb8] (rev a1)
        Subsystem: NVIDIA Corporation Device [10de:1eb8]
        Kernel driver in use: vfio-pci
        Kernel modules: nouveau
```

10.18.9.1.3. Exposer les périphériques hôtes PCI dans le cluster à l'aide de la CLI

Pour exposer les périphériques hôtes PCI dans le cluster, ajoutez des détails sur les périphériques PCI au tableau **spec.permittedHostDevices.pciHostDevices** de la ressource personnalisée (CR) **HyperConverged**.

Procédure

1. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

2. Ajoutez les informations relatives au périphérique PCI au tableau **spec.permittedHostDevices.pciHostDevices**. Par exemple :

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  permittedHostDevices: 1
    pciHostDevices: 2
```

```
    - pciDeviceSelector: "10DE:1DB6" 3
      resourceName: "nvidia.com/GV100GL_Tesla_V100" 4
    - pciDeviceSelector: "10DE:1EB8"
      resourceName: "nvidia.com/TU104GL_Tesla_T4"
    - pciDeviceSelector: "8086:6F54"
      resourceName: "intel.com/qat"
      externalResourceProvider: true 5
...
```

1. Les périphériques hôtes dont l'utilisation est autorisée dans le cluster.
2. La liste des périphériques PCI disponibles sur le nœud.
3. Le **vendor-ID** et le **device-ID** nécessaires pour identifier le dispositif PCI.
4. Le nom d'un périphérique hôte PCI.
5. Facultatif : La définition de ce champ à **true** indique que la ressource est fournie par un plugin de périphérique externe. OpenShift Virtualization permet l'utilisation de ce périphérique dans le cluster mais laisse l'allocation et la surveillance à un plugin de périphérique externe.

NOTE

L'exemple ci-dessus montre deux périphériques hôtes PCI nommés **nvidia.com/GV100GL_Tesla_V100** et **nvidia.com/TU104GL_Tesla_T4** ajoutés à la liste des périphériques hôtes autorisés dans le CR **HyperConverged**. Ces périphériques ont été testés et vérifiés pour fonctionner avec OpenShift Virtualization.

3. Enregistrez vos modifications et quittez l'éditeur.

Vérification

- Vérifiez que les périphériques hôtes PCI ont été ajoutés au nœud en exécutant la commande suivante. L'exemple de sortie montre qu'il y a un périphérique associé aux noms de ressources **nvidia.com/GV100GL_Tesla_V100**, **nvidia.com/TU104GL_Tesla_T4** et **intel.com/qat**.

```
oc describe node <node_name>
```

Exemple de sortie

```
Capacity:
  cpu:                64
  devices.kubevirt.io/kvm:        110
  devices.kubevirt.io/tun:        110
  devices.kubevirt.io/vhost-net:  110
  ephemeral-storage:              915128Mi
  hugepages-1Gi:                  0
  hugepages-2Mi:                  0
  memory:                         131395264Ki
  nvidia.com/GV100GL_Tesla_V100   1
  nvidia.com/TU104GL_Tesla_T4     1
```

```
  intel.com/qat:                1
  pods:                         250
Allocatable:
  cpu:                          63500m
  devices.kubevirt.io/kvm:      110
  devices.kubevirt.io/tun:      110
  devices.kubevirt.io/vhost-net:  110
  ephemeral-storage:            863623130526
  hugepages-1Gi:                0
  hugepages-2Mi:                0
  memory:                       130244288Ki
  nvidia.com/GV100GL_Tesla_V100   1
  nvidia.com/TU104GL_Tesla_T4     1
  intel.com/qat:                1
  pods:                         250
```

10.18.9.1.4. Suppression des périphériques hôtes PCI de la grappe à l'aide du CLI

Pour supprimer un périphérique hôte PCI du cluster, supprimez les informations relatives à ce périphérique de la ressource personnalisée (CR) **HyperConverged**.

Procédure

1. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

2. Supprimer les informations relatives au périphérique PCI du tableau **spec.permittedHostDevices.pciHostDevices** en supprimant les champs **pciDeviceSelector**, **resourceName** et **externalResourceProvider** (le cas échéant) pour le périphérique approprié. Dans cet exemple, la ressource **intel.com/qat** a été supprimée.

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  permittedHostDevices:
    pciHostDevices:
    - pciDeviceSelector: "10DE:1DB6"
      resourceName: "nvidia.com/GV100GL_Tesla_V100"
    - pciDeviceSelector: "10DE:1EB8"
      resourceName: "nvidia.com/TU104GL_Tesla_T4"
...
```

3. Enregistrez vos modifications et quittez l'éditeur.

Vérification

- Vérifiez que le périphérique hôte PCI a été supprimé du nœud en exécutant la commande

- Vérifiez que le périphérique hôte PCI a été supprimé du nœud en exécutant la commande suivante. L'exemple de sortie montre qu'il n'y a aucun périphérique associé au nom de ressource **intel.com/qat**.

```
oc describe node <node_name>
```

Exemple de sortie

```
Capacity:
  cpu:                64
  devices.kubevirt.io/kvm:        110
  devices.kubevirt.io/tun:        110
  devices.kubevirt.io/vhost-net:  110
  ephemeral-storage:              915128Mi
  hugepages-1Gi:                  0
  hugepages-2Mi:                  0
  memory:                         131395264Ki
  nvidia.com/GV100GL_Tesla_V100   1
  nvidia.com/TU104GL_Tesla_T4     1
  intel.com/qat:                  0
  pods:                           250
Allocatable:
  cpu:                63500m
  devices.kubevirt.io/kvm:        110
  devices.kubevirt.io/tun:        110
  devices.kubevirt.io/vhost-net:  110
  ephemeral-storage:              863623130526
  hugepages-1Gi:                  0
  hugepages-2Mi:                  0
  memory:                         130244288Ki
  nvidia.com/GV100GL_Tesla_V100   1
  nvidia.com/TU104GL_Tesla_T4     1
  intel.com/qat:                  0
  pods:                           250
```

10.18.9.2. Configuration des machines virtuelles pour PCI passthrough

Une fois les périphériques PCI ajoutés au cluster, vous pouvez les affecter aux machines virtuelles. Les périphériques PCI sont désormais disponibles comme s'ils étaient physiquement connectés aux machines virtuelles.

10.18.9.2.1. Attribution d'un périphérique PCI à une machine virtuelle

Lorsqu'un périphérique PCI est disponible dans un cluster, vous pouvez l'affecter à une machine virtuelle et activer le PCI passthrough.

Procédure

- Attribuer le périphérique PCI à une machine virtuelle en tant que périphérique hôte.

Exemple :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
```

```
spec:
  domain:
    devices:
      hostDevices:
      - deviceName: nvidia.com/TU104GL_Tesla_T4 1
        name: hostdevices1
```

1 Le nom du périphérique PCI autorisé sur le cluster en tant que périphérique hôte. La machine virtuelle peut accéder à ce périphérique hôte.

Vérification

- Utilisez la commande suivante pour vérifier que le périphérique hôte est disponible dans la machine virtuelle.

```
$ lspci -nnk | grep NVIDIA
```

Exemple de sortie

```
$ 02:01.0 3D controller [0302]: NVIDIA Corporation GV100GL [Tesla V100 PCIe 32GB]
[10de:1eb8] (rev a1)
```

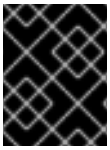
10.18.9.3. Ressources supplémentaires

- Activation des extensions matérielles de virtualisation Intel VT-X et AMD-V dans le BIOS
- Gestion des autorisations de fichiers
- Tâches de configuration de la machine après l'installation

10.18.10. Configuration du vGPU passthrough

Vos machines virtuelles peuvent accéder à un matériel GPU virtuel (vGPU). L'attribution d'un vGPU à votre machine virtuelle vous permet d'effectuer les opérations suivantes :

- Accédez à une fraction du GPU du matériel sous-jacent pour obtenir des performances élevées dans votre machine virtuelle.
- Rationaliser les opérations d'E/S gourmandes en ressources.



IMPORTANT

la fonction vGPU passthrough ne peut être attribuée qu'à des appareils connectés à des clusters fonctionnant dans un environnement "bare metal".

10.18.10.1. Attribution de périphériques vGPU passthrough à une machine virtuelle

Utilisez la console web d'OpenShift Container Platform pour attribuer des périphériques vGPU passthrough à votre machine virtuelle.

Conditions préalables

- La machine virtuelle doit être arrêtée.

Procédure

1. Dans la console web de OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez la machine virtuelle à laquelle vous souhaitez attribuer le périphérique.
3. Dans l'onglet **Details**, cliquez sur **GPU devices**.
   Si vous ajoutez un périphérique vGPU en tant que périphérique hôte, vous ne pouvez pas accéder au périphérique avec la console VNC.
4. Cliquez sur **Add GPU device**, entrez dans **Name** et sélectionnez l'appareil dans la liste **Device name**.
5. Cliquez sur **Save**.
6. Cliquez sur l'onglet **YAML** pour vérifier que les nouveaux appareils ont été ajoutés à la configuration de votre cluster dans la section **hostDevices**.

NOTE

Vous pouvez ajouter des périphériques matériels aux machines virtuelles créées à partir de modèles personnalisés ou d'un fichier YAML. Vous ne pouvez pas ajouter de périphériques à des modèles de source de démarrage fournis à l'avance pour des systèmes d'exploitation spécifiques, tels que Windows 10 ou RHEL 7.

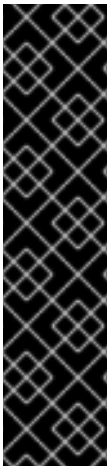
Pour afficher les ressources connectées à votre cluster, cliquez sur **Compute → Hardware Devices** dans le menu latéral.

10.18.10.2. Ressources supplémentaires

- Création de machines virtuelles
- Création de modèles de machines virtuelles

10.18.11. Configuration des dispositifs à médiation

OpenShift Virtualization crée automatiquement des périphériques médiatisés, tels que des GPU virtuels (vGPU), si vous fournissez une liste de périphériques dans la ressource personnalisée (CR) **HyperConverged**.



IMPORTANT

La configuration déclarative des périphériques médiatisés est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes d'un point de vue fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, permettant aux clients de tester les fonctionnalités et de fournir un retour d'information au cours du processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

10.18.11.1. A propos de l'utilisation de NVIDIA GPU Operator

Le NVIDIA GPU Operator gère les ressources GPU NVIDIA dans un cluster OpenShift Container Platform et automatise les tâches liées au démarrage des nœuds GPU. Étant donné que le GPU est une ressource spéciale dans le cluster, vous devez installer certains composants avant de déployer des charges de travail d'application sur le GPU. Ces composants comprennent les pilotes NVIDIA qui activent l'architecture CUDA (compute unified device architecture), le plugin de périphérique Kubernetes, le runtime de conteneur et d'autres éléments tels que l'étiquetage automatique des nœuds, la surveillance et plus encore.

NOTE

NVIDIA GPU Operator n'est pris en charge que par NVIDIA. Pour plus d'informations sur l'obtention d'un support de la part de NVIDIA, voir Obtenir un support de la part de NVIDIA.

Il existe deux façons d'activer les GPU avec OpenShift Container Platform OpenShift Virtualization : la méthode native d'OpenShift Container Platform décrite ici et l'utilisation de NVIDIA GPU Operator.

L'opérateur NVIDIA GPU est un opérateur Kubernetes qui permet à OpenShift Container Platform OpenShift Virtualization d'exposer les GPU aux charges de travail virtualisées fonctionnant sur OpenShift Container Platform. Il permet aux utilisateurs de provisionner et de gérer facilement des machines virtuelles dotées de GPU, en leur donnant la possibilité d'exécuter des charges de travail complexes d'intelligence artificielle/apprentissage machine (AI/ML) sur la même plateforme que leurs autres charges de travail. Il offre également un moyen simple de faire évoluer la capacité GPU de leur infrastructure, permettant ainsi une croissance rapide des charges de travail basées sur le GPU.

Pour plus d'informations sur l'utilisation de NVIDIA GPU Operator pour provisionner des nœuds de travail pour l'exécution de VM accélérées par le GPU, voir NVIDIA GPU Operator avec OpenShift Virtualization.

10.18.11.2. À propos de l'utilisation de GPU virtuels avec OpenShift Virtualization

Certaines cartes de processeurs graphiques (GPU) prennent en charge la création de GPU virtuels (vGPU). OpenShift Virtualization peut créer automatiquement des vGPU et d'autres périphériques médiatisés si un administrateur fournit des détails de configuration dans la ressource personnalisée (CR) **HyperConverged**. Cette automatisation est particulièrement utile pour les grands clusters.

NOTE

Reportez-vous à la documentation de votre fournisseur de matériel pour plus de détails sur les fonctionnalités et l'assistance.

Dispositif de médiation

Un dispositif physique divisé en un ou plusieurs dispositifs virtuels. Un vGPU est un type de périphérique médiatisé (mdev) ; les performances du GPU physique sont réparties entre les périphériques virtuels. Vous pouvez attribuer des périphériques médiatisés à une ou plusieurs machines virtuelles (VM), mais le nombre d'invités doit être compatible avec votre GPU. Certains GPU ne prennent pas en charge plusieurs invités.

10.18.11.2.1. Conditions préalables

- Si votre fournisseur de matériel fournit des pilotes, vous les avez installés sur les nœuds où vous souhaitez créer des dispositifs à médiation.
  - Si vous utilisez des cartes NVIDIA, vous avez installé le pilote NVIDIA GRID .

10.18.11.2.2. Aperçu de la configuration

Lors de la configuration des dispositifs à médiation, l'administrateur doit effectuer les tâches suivantes :

- Créer les dispositifs de médiation.
- Exposer les dispositifs médiatisés au cluster.

La CR **HyperConverged** comprend des API qui permettent d'accomplir ces deux tâches.

Créer des dispositifs médiatisés

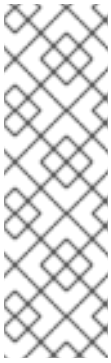
```
...
spec:
  mediatedDevicesConfiguration:
    mediatedDevicesTypes: 1
    - <device_type>
    nodeMediatedDeviceTypes: 2
    - mediatedDevicesTypes: 3
      - <device_type>
      nodeSelector: 4
        <node_selector_key>: <node_selector_value>
...
```

1. Obligatoire : Configure les paramètres globaux du cluster.
2. Facultatif : Remplace la configuration globale pour un nœud ou un groupe de nœuds spécifique. Doit être utilisé avec la configuration globale **mediatedDevicesTypes**.
3. Obligatoire si vous utilisez **nodeMediatedDeviceTypes**. Remplace la configuration globale de **mediatedDevicesTypes** pour les nœuds spécifiés.
4. Requis si vous utilisez **nodeMediatedDeviceTypes**. Doit inclure une paire **key:value**.

Exposer les dispositifs à médiation au cluster

```
...
  permittedHostDevices:
    mediatedDevices:
    - mdevNameSelector: GRID T4-2Q 1
      resourceName: nvidia.com/GRID_T4-2Q 2
...
```

1 Expose les dispositifs médiatisés qui correspondent à cette valeur sur l'hôte.



NOTE

Vous pouvez voir les types de dispositifs médiatisés pris en charge par votre dispositif en consultant le contenu de **/sys/bus/pci/devices/<slot>:<bus>:<domain>.<function>/mdev_supported_types/<type>/name**, en remplaçant les valeurs correctes pour votre système.

Par exemple, le fichier de noms du type **nvidia-231** contient la chaîne de sélection **GRID T4-2Q**. L'utilisation de **GRID T4-2Q** comme valeur de **mdevNameSelector** permet aux nœuds d'utiliser le type **nvidia-231**.

2 Le **resourceName** doit correspondre à celui alloué sur le nœud. Trouvez le **resourceName** en utilisant la commande suivante :

```
$ oc get $NODE -o json  \
| jq '.status.allocatable | \
with_entries(select(.key | startswith("nvidia.com/"))) | \
with_entries(select(.value != "0"))'
```

10.18.11.2.3. Comment les vGPU sont affectés aux nœuds

Pour chaque appareil physique, OpenShift Virtualization configure les valeurs suivantes :

- Un seul type de mdev.
- Nombre maximal d'instances du type **mdev** sélectionné.

L'architecture de la grappe affecte la manière dont les dispositifs sont créés et attribués aux nœuds.

Grand cluster avec plusieurs cartes par nœud

Sur les nœuds dotés de plusieurs cartes pouvant prendre en charge des types de vGPU similaires, les types de périphériques concernés sont créés à la ronde. Par exemple :

```
...
mediatedDevicesConfiguration:
  mediatedDevicesTypes:
  - nvidia-222
  - nvidia-228
  - nvidia-105
  - nvidia-108
...
```

Dans ce scénario, chaque nœud dispose de deux cartes, toutes deux compatibles avec les types de vGPU suivants :

```
nvidia-105
...
nvidia-108
nvidia-217
nvidia-299
...
```

Sur chaque nœud, OpenShift Virtualization crée les vGPU suivants :

- 16 vGPUs de type nvidia-105 sur la première carte.
- 2 vGPUs de type nvidia-108 sur la seconde carte.

Un nœud possède une seule carte qui prend en charge plusieurs types de vGPU

OpenShift Virtualization utilise le type pris en charge qui vient en premier sur la liste **mediatedDevicesTypes**.
Par exemple, la carte d'un nœud prend en charge **nvidia-223** et **nvidia-224**. La liste **mediatedDevicesTypes** suivante est configurée :

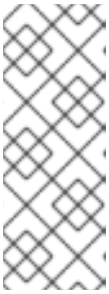
```
...
mediatedDevicesConfiguration:
  mediatedDevicesTypes:
  - nvidia-22
  - nvidia-223
  - nvidia-224
...
```

Dans cet exemple, OpenShift Virtualization utilise le type **nvidia-223**.

10.18.11.2.4. Sur la modification et la suppression des dispositifs médiatisés

La configuration du dispositif médiatisé du cluster peut être mise à jour avec OpenShift Virtualization en :

- Editer le **HyperConverged** CR et modifier le contenu de la strophe **mediatedDevicesTypes**.
- Modification des étiquettes de nœuds correspondant au sélecteur de nœuds **nodeMediatedDeviceTypes**.
- Suppression des informations relatives au dispositif dans les strophes **spec.mediatedDevicesConfiguration** et **spec.permittedHostDevices** de la CR **HyperConverged**.



NOTE

Si vous supprimez les informations relatives au dispositif de la strophe **spec.permittedHostDevices** sans les supprimer également de la strophe **spec.mediatedDevicesConfiguration**, vous ne pouvez pas créer un nouveau type de dispositif à médiation sur le même nœud. Pour supprimer correctement les dispositifs à médiation, supprimez les informations relatives aux dispositifs dans les deux strophes.

En fonction des changements spécifiques, ces actions amènent OpenShift Virtualization à reconfigurer les périphériques médiatisés ou à les supprimer des nœuds du cluster.

10.18.11.2.5. Préparation des hôtes pour les dispositifs à médiation

Vous devez activer le pilote Input-Output Memory Management Unit (IOMMU) avant de pouvoir configurer les périphériques à médiation.

10.18.11.2.5.1. Ajout d'arguments au noyau pour activer le pilote IOMMU

Pour activer le pilote IOMMU (Input-Output Memory Management Unit) dans le noyau, créez l'objet **MachineConfig** et ajoutez les arguments du noyau.

Conditions préalables

- Privilège administratif sur un cluster OpenShift Container Platform en fonctionnement.
- Matériel CPU Intel ou AMD.
- La technologie de virtualisation Intel pour les extensions Directed I/O ou AMD IOMMU dans le BIOS (Basic Input/Output System) est activée.

Procédure

1. Créez un objet **MachineConfig** qui identifie l'argument du noyau. L'exemple suivant montre un argument du noyau pour un processeur Intel.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker 1
  name: 100-worker-iommu 2
spec:
  config:
    ignition:
      version: 3.2.0
  kernelArguments:
      - intel_iommu=on 3
...
```

1. Applique le nouvel argument du noyau uniquement aux nœuds de travail.
2. L'adresse **name** indique le rang de cet argument du noyau (100) parmi les configurations de la machine et son objectif. Si vous avez un processeur AMD, spécifiez l'argument du noyau comme **amd_iommu=on**.
3. Identifie l'argument du noyau comme étant **intel_iommu** pour un processeur Intel.

2. Créer le nouvel objet **MachineConfig**:

```
$ oc create -f 100-worker-kernel-arg-iommu.yaml
```

Vérification

- Vérifiez que le nouvel objet **MachineConfig** a bien été ajouté.

```
$ oc get MachineConfig
```

10.18.11.2.6. Ajout et suppression de dispositifs médiatisés

Vous pouvez ajouter ou supprimer des dispositifs médiatisés.

10.18.11.2.6.1. Créer et exposer des dispositifs médiatisés

Vous pouvez exposer et créer des dispositifs médiatisés tels que des GPU virtuels (vGPU) en modifiant la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- Vous avez activé le pilote IOMMU (Input-Output Memory Management Unit).

Procédure

1. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

2. Ajoutez les informations sur le dispositif médiatisé à la CR **HyperConverged spec** , en veillant à inclure les strophes **mediatedDevicesConfiguration** et **permittedHostDevices**. Par exemple :

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  mediatedDevicesConfiguration: <.>
    mediatedDevicesTypes: <.>
    - nvidia-231
    nodeMediatedDeviceTypes: <.>
    - mediatedDevicesTypes: <.>
      - nvidia-233
      nodeSelector:
        kubernetes.io/hostname: node-11.redhat.com
  permittedHostDevices: <.>
    mediatedDevices:
    - mdevNameSelector: GRID T4-2Q
      resourceName: nvidia.com/GRID_T4-2Q
    - mdevNameSelector: GRID T4-8Q
      resourceName: nvidia.com/GRID_T4-8Q
...
```

<.> Crée des dispositifs médiatisés. <.> Requis : Configuration globale **mediatedDevicesTypes**. <.> Facultatif : Remplace la configuration globale pour des nœuds spécifiques. <.> Requis si vous utilisez **nodeMediatedDeviceTypes**<.> Expose les périphériques à médiation au cluster.

3. Enregistrez vos modifications et quittez l'éditeur.

Vérification

- Vous pouvez vérifier qu'un dispositif a été ajouté à un nœud spécifique en exécutant la commande suivante :

```
oc describe node <node_name>
```

10.18.11.2.6.2. Suppression des dispositifs à médiation de la grappe à l'aide de la CLI

Pour supprimer un dispositif médiatisé du cluster, supprimez les informations relatives à ce dispositif de la ressource personnalisée (CR) **HyperConverged**.

Procédure

1. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

2. Supprimez les informations relatives au dispositif dans les strophes **spec.mediatedDevicesConfiguration** et **spec.permittedHostDevices** de la CR **HyperConverged**. La suppression des deux entrées garantit la possibilité de créer ultérieurement un nouveau type de dispositif à médiation sur le même nœud. Par exemple :

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  mediatedDevicesConfiguration:
    mediatedDevicesTypes: 1
      - nvidia-231
  permittedHostDevices:
    mediatedDevices: 2
    - mdevNameSelector: GRID T4-2Q
      resourceName: nvidia.com/GRID_T4-2Q
```

1. Pour supprimer le type de dispositif **nvidia-231**, supprimez-le du tableau **mediatedDevicesTypes**.

2. Pour supprimer le dispositif **GRID T4-2Q**, supprimez le champ **mdevNameSelector** et le champ **resourceName** correspondant.

3. Enregistrez vos modifications et quittez l'éditeur.

10.18.11.3. Utilisation de dispositifs médiatisés

Un vGPU est un type de dispositif médiatisé ; les performances du GPU physique sont réparties entre les dispositifs virtuels. Vous pouvez attribuer des périphériques médiatisés à une ou plusieurs machines virtuelles.

10.18.11.3.1. Attribution d'un dispositif médiatisé à une machine virtuelle

Attribuer des périphériques médiatisés tels que des GPU virtuels (vGPU) à des machines virtuelles.

Conditions préalables

- Le dispositif médiatisé est configuré dans la ressource personnalisée **HyperConverged**.

Procédure

- Attribuez le périphérique médiatisé à une machine virtuelle (VM) en modifiant la strophe **spec.domain.devices.gpus** du manifeste **VirtualMachine**:

  Exemple de manifeste de machine virtuelle

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  domain:
    devices:
      gpus:
      - deviceName: nvidia.com/TU104GL_Tesla_T4 1
        name: gpu1 2
      - deviceName: nvidia.com/GRID_T4-1Q
        name: gpu2
```

1. Le nom de la ressource associée au dispositif médiatisé.
2. Nom permettant d'identifier le périphérique sur la VM.

Vérification

- Pour vérifier que le périphérique est disponible dans la machine virtuelle, exécutez la commande suivante, en remplaçant **<device_name>** par la valeur **deviceName** du manifeste **VirtualMachine**:

```
$ lspci -nnk | grep <device_name>
```

10.18.11.4. Ressources supplémentaires

- Activation des extensions matérielles de virtualisation Intel VT-X et AMD-V dans le BIOS

10.18.12. Configuration d'un chien de garde

Exposer un chien de garde en configurant la machine virtuelle (VM) pour un périphérique de chien de garde, en installant le chien de garde et en démarrant le service de chien de garde.

10.18.12.1. Conditions préalables

- La machine virtuelle doit disposer d'une prise en charge par le noyau d'un périphérique de surveillance **i6300esb**. Les images Red Hat Enterprise Linux (RHEL) prennent en charge **i6300esb**.

10.18.12.2. Définition d'un dispositif de surveillance

Définir comment le chien de garde procède lorsque le système d'exploitation (OS) ne répond plus.

Tableau 10.4. Actions disponibles

poweroff	La machine virtuelle (VM) s'éteint immédiatement. Si **spec.running** est défini sur **true**, ou si **spec.runStrategy** n'est pas défini sur **manual**, la VM redémarre.
reset	La VM redémarre sur place et le système d'exploitation invité ne peut pas réagir. Étant donné que le temps nécessaire au redémarrage du système d'exploitation invité peut entraîner un dépassement du délai d'exécution des sondes d'intégrité, l'utilisation de cette option est déconseillée. Ce délai peut prolonger le temps nécessaire au redémarrage de la VM si les protections au niveau du cluster remarquent que la sonde d'intégrité a échoué et la replanifient de force.
shutdown	La VM s'éteint de manière gracieuse en arrêtant tous les services.

Procédure

1. Créez un fichier YAML avec le contenu suivant :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    kubevirt.io/vm: vm2-rhel84-watchdog
  name: <vm-name>
spec:
  running: false
  template:
    metadata:
     labels:
       kubevirt.io/vm: vm2-rhel84-watchdog
    spec:
      domain:
        devices:
          watchdog:
            name: <watchdog>
            i6300esb:
              action: "poweroff" 1
...
```

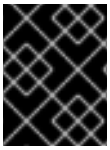
1 Spécifiez l'action **watchdog** (**poweroff**, **reset**, ou **shutdown**).

L'exemple ci-dessus configure le dispositif de surveillance **i6300esb** sur une VM RHEL8 avec l'action de mise hors tension et expose le dispositif en tant que /**dev**/**watchdog**.

Ce dispositif peut maintenant être utilisé par le binaire chien de garde.

2. Appliquez le fichier YAML à votre cluster en exécutant la commande suivante :

```
oc apply -f <nom_du_fichier>.yaml
```



IMPORTANT

Cette procédure sert uniquement à tester la fonctionnalité du chien de garde et ne doit pas être exécutée sur des machines de production.

1. Exécutez la commande suivante pour vérifier que la VM est connectée au dispositif de surveillance :

```
$ lspci | grep watchdog -i
```

2. Exécutez l'une des commandes suivantes pour confirmer que le chien de garde est actif :

- Déclencher une panique du noyau :

```
# echo c > /proc/sysrq-trigger
```

- Mettre fin au service de chien de garde :

```
# pkill -9 watchdog
```

10.18.12.3. Installation d'un dispositif de surveillance

Installez le paquet **watchdog** sur votre machine virtuelle et démarrez le service watchdog.

Procédure

1. En tant qu'utilisateur root, installez le paquetage **watchdog** et ses dépendances :

```
# yum install watchdog
```

2. Décommentez la ligne suivante dans le fichier **/etc/watchdog.conf** et enregistrez les modifications :

```
#watchdog-device = /dev/watchdog
```

3. Activer le service watchdog pour qu'il démarre au démarrage :

```
# systemctl enable --now watchdog.service
```

10.18.12.4. Ressources supplémentaires

- Contrôler l'état de santé des applications à l'aide de bilans de santé

10.18.13. Importation et mise à jour automatiques de sources de démarrage prédéfinies

Vous pouvez utiliser des sources de démarrage qui sont *system-defined* et incluses avec OpenShift Virtualization ou *user-defined*, que vous créez. Les importations et les mises à jour de sources de démarrage définies par le système sont contrôlées par le portail de fonctionnalités du produit. Vous pouvez activer, désactiver ou réactiver les mises à jour à l'aide du portail de fonctionnalités. Les sources de démarrage définies par l'utilisateur ne sont pas contrôlées par le portail de fonctionnalités du produit et doivent être gérées individuellement pour activer ou désactiver les importations et les mises à jour automatiques.

IMPORTANT

À partir de la version 4.10, OpenShift Virtualization importe et met à jour automatiquement les sources de démarrage, sauf si vous vous désengagez manuellement ou si vous ne définissez pas de classe de stockage par défaut.

Si vous passez à la version 4.10, vous devez activer manuellement les importations et les mises à jour automatiques pour les sources de démarrage de la version 4.9 ou antérieure.

10.18.13.1. Activation des mises à jour automatiques des sources de démarrage

Si vous avez des sources de démarrage d'OpenShift Virtualization 4.9 ou antérieures, vous devez activer manuellement les mises à jour automatiques pour ces sources de démarrage. Toutes les sources de démarrage d'OpenShift Virtualization 4.10 et plus sont automatiquement mises à jour par défaut.

Pour activer les importations et les mises à jour automatiques des sources de démarrage, définissez le champ **cdi.kubevirt.io/dataImportCron** sur **true** pour chaque source de démarrage que vous souhaitez mettre à jour automatiquement.

Procédure

- Pour activer les mises à jour automatiques pour une source de démarrage, utilisez la commande suivante pour appliquer l'étiquette **dataImportCron** à la source de données :

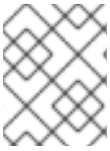
```
oc label --overwrite DataSource rhel8 -n openshift-virtualization-os-images
cdi.kubevirt.io/dataImportCron=true (1)
```

(1) En spécifiant **true**, vous activez les mises à jour automatiques pour la source de démarrage **rhel8**.

10.18.13.2. Désactivation des mises à jour automatiques des sources de démarrage

La désactivation des importations et des mises à jour automatiques de la source de démarrage peut s'avérer utile pour réduire le nombre de journaux dans les environnements déconnectés ou pour réduire l'utilisation des ressources.

Pour désactiver les importations et les mises à jour automatiques de la source de démarrage, définissez le champ **spec.featureGates.enableCommonBootImageImport** dans la ressource personnalisée (CR) **HyperConverged** sur **false**.



NOTE

Les sources de démarrage définies par l'utilisateur ne sont pas affectées par ce paramètre.

Procédure

- Utilisez la commande suivante pour désactiver les mises à jour automatiques de la source de démarrage :

```
$ oc patch hco kubevirt-hyperconverged -n openshift-cnv \
 --type json -p '[{"op": "replace", "path":
"/spec/featureGates/enableCommonBootImageImport", \
 "value": false}]'
```

10.18.13.3. Réactivation des mises à jour automatiques des sources de démarrage

Si vous avez précédemment désactivé les mises à jour automatiques des sources de démarrage, vous devez réactiver manuellement cette fonctionnalité. Définissez le champ **spec.featureGates.enableCommonBootImageImport** dans la ressource personnalisée (CR) **HyperConverged** sur **true**.

Procédure

- Utilisez la commande suivante pour réactiver les mises à jour automatiques :

```
$ oc patch hco kubevirt-hyperconverged -n openshift-cnv --type json -p '[{"op": "replace",
"path": "/spec/featureGates/enableCommonBootImageImport", "value": true}]'
```

10.18.13.4. Configuration d'une classe de stockage pour les mises à jour de sources de démarrage définies par l'utilisateur

Vous pouvez configurer une classe de stockage qui permet l'importation et la mise à jour automatiques des sources de démarrage définies par l'utilisateur.

Procédure

1. Définir un nouveau **storageClassName** en modifiant la ressource personnalisée (CR) **HyperConverged**.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  dataImportCronTemplates:
  - metadata:
      name: rhel8-image-cron
    spec:
      template:
        spec:
          storageClassName: <appropriate_class_name>
...
```

2. Définissez la nouvelle classe de stockage par défaut en exécutant les commandes suivantes :

```
$ oc patch storageclass <current_default_storage_class> -p '{"metadata" : {"annotations":
{"storageclass.kubernetes.io/is-default-class":\N "false"}}}'
```

```
$ oc patch storageclass <appropriate_storage_class> -p '{"metadata" : {"annotations":
{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

10.18.13.5. Activation des mises à jour automatiques pour les sources de démarrage définies par l'utilisateur

OpenShift Virtualization met automatiquement à jour les sources de démarrage définies par le système par défaut, mais ne met pas automatiquement à jour les sources de démarrage définies par l'utilisateur. Vous devez activer manuellement les importations et les mises à jour automatiques sur les sources de démarrage définies par l'utilisateur en modifiant la ressource personnalisée (CR) **HyperConverged**.

Procédure

1. La commande suivante permet d'ouvrir le CR **HyperConverged** pour l'éditer :

```
$ oc edit -n openshift-cnv HyperConverged
```

2. Modifiez le CR **HyperConverged** en ajoutant le modèle et la source de démarrage appropriés dans la section **dataImportCronTemplates**. Par exemple :

Exemple sous CentOS 7

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  dataImportCronTemplates:
  - metadata:
      name: centos7-image-cron
      annotations:
        cdi.kubevirt.io/storage.bind.immediate.requested: "true" 1
    spec:
      schedule: "0 */12 * * *" 2
      template:
        spec:
          source:
            registry: 3
              url: docker://quay.io/containerdisks/centos:7-2009
          storage:
            resources:
              requests:
                storage: 10Gi
      managedDataSource: centos7 4
      retentionPolicy: "None" 5
```

1 Cette annotation est requise pour les classes de stockage dont la valeur **volumeBindingMode** est fixée à **WaitForFirstConsumer**.

2 Planification du travail spécifié au format cron.

3 Permet de créer un volume de données à partir d'une source de registre. Utilisez la valeur par défaut **pod pullMethod** et non **node pullMethod** , qui est basée sur le cache de docker **node**. Le cache docker **node** est utile lorsqu'une image de registre est disponible via **Container.Image**, mais que l'importateur CDI n'est pas autorisé à y accéder.

4 Pour que l'image personnalisée soit détectée comme source de démarrage disponible, le nom de l'image **managedDataSource** doit correspondre au nom du modèle **DataSource**, qui se trouve sous **spec.dataVolumeTemplates.spec.sourceRef.name** dans le fichier YAML du modèle VM.

5 Utilisez **All** pour conserver les volumes et les sources de données lorsque la tâche cron est supprimée. Utilisez **None** pour supprimer les volumes et les sources de données lorsque la tâche cron est supprimée.

10.18.13.6. Désactivation d'une mise à jour automatique pour une source de démarrage définie par le système ou par l'utilisateur

Vous pouvez désactiver les importations et les mises à jour automatiques pour une source de démarrage définie par l'utilisateur et pour une source de démarrage définie par le système.

Les sources de démarrage définies par le système n'étant pas répertoriées par défaut dans le site **spec.dataImportCronTemplates** de la ressource personnalisée (CR) **HyperConverged**, vous devez ajouter la source de démarrage et désactiver les importations et mises à jour automatiques.

Procédure

- Pour désactiver les importations et les mises à jour automatiques d'une source de démarrage définie par l'utilisateur, supprimez la source de démarrage du champ **spec.dataImportCronTemplates** dans la liste des ressources personnalisées.
- Pour désactiver les importations et les mises à jour automatiques d'une source de démarrage définie par le système :
  - Modifiez le CR **HyperConverged** et ajoutez la source de démarrage à **spec.dataImportCronTemplates**.
  - Désactivez les importations et les mises à jour automatiques en définissant l'annotation **dataimportcrontemplate.kubevirt.io/enable** sur **false**. Par exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  dataImportCronTemplates:
  - metadata:
      annotations:
        dataimportcrontemplate.kubevirt.io/enable: false
      name: rhel8-image-cron
...
```

10.18.13.7. Vérification de l'état d'une source de démarrage

Vous pouvez vérifier si une source de démarrage est définie par le système ou par l'utilisateur.

La section **status** de chaque source d'amorçage répertoriée dans le champ **status.dataImportChronTemplates** du CR **HyperConverged** indique le type de source d'amorçage. Par exemple, **commonTemplate: true** indique une source d'amorçage définie par le système (**commonTemplate**) et **status: {}** indique une source d'amorçage définie par l'utilisateur.

Procédure

1. Utilisez la commande **oc get** pour dresser la liste des **dataImportChronTemplates** dans le CR **HyperConverged**.

2. Vérifier l'état de la source de démarrage.

Exemple de sortie

```
...
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
...
spec:
  ...
status: 1
  ...
  dataImportCronTemplates: 2
  - metadata:
      annotations:
        cdi.kubevirt.io/storage.bind.immediate.requested: "true"
      name: centos-7-image-cron
    spec:
      garbageCollect: Outdated
      managedDataSource: centos7
      schedule: 55 8/12 * * *
      template:
        metadata: {}
        spec:
          source:
            registry:
              url: docker://quay.io/containerdisks/centos:7-2009
          storage:
            resources:
              requests:
                storage: 30Gi
      status: {}
    status:
      commonTemplate: true 3
    ...
  - metadata:
      annotations:
        cdi.kubevirt.io/storage.bind.immediate.requested: "true"
      name: user-defined-dic
    spec:
      garbageCollect: Outdated
      managedDataSource: user-defined-centos-stream8
      schedule: 55 8/12 * * *
```

```
    template:
      metadata: {}
      spec:
        source:
          registry:
            pullMethod: node
            url: docker://quay.io/containerdisks/centos-stream:8
        storage:
          resources:
            requests:
              storage: 30Gi
      status: {}
    status: {} 4
...
```

1. Le champ **status** pour le CR **HyperConverged**.
2. Le champ **dataImportCronTemplates**, qui répertorie toutes les sources de démarrage définies.
3. Indique une source de démarrage définie par le système.
4. Indique une source de démarrage définie par l'utilisateur.

10.18.14. Activation des évictions du désordre sur les machines virtuelles

Vous pouvez utiliser l'ordonnanceur pour expulser les pods afin qu'ils puissent être replanifiés sur des nœuds plus appropriés. Si le pod est une machine virtuelle, l'éviction du pod entraîne la migration en direct de la machine virtuelle vers un autre nœud.

IMPORTANT

L'éviction du Descheduler pour les machines virtuelles est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

10.18.14.1. Profils du déschedulateur

Utilisez le profil Technology Preview **DevPreviewLongLifecycle** pour activer le descheduler sur une machine virtuelle. Il s'agit du seul profil de déscheduler actuellement disponible pour OpenShift Virtualization. Pour garantir une planification correcte, créez des VM avec des demandes de CPU et de mémoire pour la charge attendue.

DevPreviewLongLifecycle

Ce profil équilibre l'utilisation des ressources entre les nœuds et permet les stratégies suivantes :

- **RemovePodsHavingTooManyRestarts**: supprime les pods dont les conteneurs ont été redémarrés trop de fois et les pods dont la somme des redémarrages de tous les conteneurs (y compris les Init Containers) est supérieure à 100. Le redémarrage du système d'exploitation invité de la VM n'augmente pas ce nombre.

- **LowNodeUtilization**l'ordonnanceur : évince les pods des nœuds sur-utilisés lorsqu'il y a des nœuds sous-utilisés. Le nœud de destination du pod évincé sera déterminé par l'ordonnanceur.

    - Un nœud est considéré comme sous-utilisé si son utilisation est inférieure à 20 ou à tous les seuils (CPU, mémoire et nombre de pods).

    - Un nœud est considéré comme surutilisé si son utilisation est supérieure à 50 ou à l'un des seuils (CPU, mémoire et nombre de pods).

10.18.14.2. Installation du déschedulateur

Le descheduler n'est pas disponible par défaut. Pour l'activer, vous devez installer Kube Descheduler Operator depuis OperatorHub et activer un ou plusieurs profils de descheduler.

Par défaut, le descheduler fonctionne en mode prédictif, ce qui signifie qu'il ne fait que simuler les évictions de pods. Vous devez changer le mode en mode automatique pour que le descheduler effectue les évictions de pods.

IMPORTANT

Si vous avez activé les plans de contrôle hébergés dans votre cluster, définissez un seuil de priorité personnalisé pour réduire le risque d'éviction des pods dans les espaces de noms des plans de contrôle hébergés. Définissez le nom de la classe de seuil de priorité sur **hypershift-control-plane**, car elle a la valeur de priorité la plus basse (**100000000**) des classes de priorité du plan de contrôle hébergé.

Conditions préalables

- Privilèges d'administrateur de cluster.
- Accès à la console web d'OpenShift Container Platform.

Procédure

1. Connectez-vous à la console web de OpenShift Container Platform.
2. Créer l'espace de noms requis pour l'opérateur Kube Descheduler.
    a. Naviguez jusqu'à **Administration → Namespaces** et cliquez sur **Create Namespace**.
    b. Entrez **openshift-kube-descheduler-operator** dans le champ **Name**, entrez **openshift.io/cluster-monitoring=true** dans le champ **Labels** pour activer les métriques du déscheduler, et cliquez sur **Create**.
3. Installez l'opérateur Kube Descheduler.
    a. Naviguez jusqu'à **Operators → OperatorHub**.
    b. Tapez **Kube Descheduler Operator** dans le champ de filtre.

c. Sélectionnez le site **Kube Descheduler Operator** et cliquez sur **Install**.

d. Sur la page **Install Operator**, sélectionnez **A specific namespace on the cluster**. Sélectionnez **openshift-kube-descheduler-operator** dans le menu déroulant.

e. Ajustez les valeurs de **Update Channel** et **Approval Strategy** aux valeurs souhaitées.

f. Cliquez sur **Install**.

4. Créer une instance de déscheduler.

a. Dans la page **Operators → Installed Operators**, cliquez sur **Kube Descheduler Operator**.

b. Sélectionnez l'onglet **Kube Descheduler** et cliquez sur **Create KubeDescheduler**.

c. Modifiez les paramètres si nécessaire.

i. Pour expulser des pods au lieu de simuler les expulsions, remplacez le champ **Mode** par **Automatic**.

ii. Développez la section **Profiles** et sélectionnez **DevPreviewLongLifecycle**. Le profil **AffinityAndTaints** est activé par défaut.

IMPORTANT

Le seul profil actuellement disponible pour OpenShift Virtualization est **DevPreviewLongLifecycle**.

Vous pouvez également configurer les profils et les paramètres du déscheduler ultérieurement à l'aide de la CLI OpenShift (**oc**).

10.18.14.3. Activation des évictions du déscheduler sur une machine virtuelle (VM)

Une fois le descheduler installé, vous pouvez activer les évictions du descheduler sur votre VM en ajoutant une annotation à la ressource personnalisée (CR) **VirtualMachine**.

Conditions préalables

- Installez le descheduler dans la console web de OpenShift Container Platform ou OpenShift CLI (**oc**).
- Assurez-vous que la machine virtuelle n'est pas en cours d'exécution.

Procédure

1. Avant de démarrer la VM, ajoutez l'annotation **descheduler.alpha.kubernetes.io/evict** au CR **VirtualMachine**:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  template:
    metadata:
      annotations:
        descheduler.alpha.kubernetes.io/evict: "true"
```

2. Si vous n'avez pas déjà défini le profil **DevPreviewLongLifecycle** dans la console web lors de l'installation, spécifiez le profil **DevPreviewLongLifecycle** dans la section **spec.profile** de l'objet **KubeDescheduler**:

```
apiVersion: operator.openshift.io/v1
kind: KubeDescheduler
metadata:
  name: cluster
  namespace: openshift-kube-descheduler-operator
spec:
  deschedulingIntervalSeconds: 3600
  profiles:
  - DevPreviewLongLifecycle
  mode: Predictive (1)
```

(1) Par défaut, le descheduler n'évince pas les pods. Pour évincer des pods, définissez **mode** à **Automatic**.

L'ordonnanceur est maintenant activé sur la VM.

10.18.14.4. Ressources supplémentaires

- Éviction des pods à l'aide de l'ordonnanceur

10.19. IMPORTER DES MACHINES VIRTUELLES

10.19.1. Certificats TLS pour l'importation de volumes de données

10.19.1.1. Ajout de certificats TLS pour l'authentification des importations de volumes de données

Les certificats TLS pour le registre ou les points de terminaison HTTPS doivent être ajoutés à une carte de configuration pour importer des données à partir de ces sources. Cette carte de configuration doit être présente dans l'espace de noms du volume de données de destination.

Créez la carte de configuration en référençant le chemin d'accès relatif au certificat TLS.

Procédure

1. Assurez-vous que vous êtes dans le bon espace de noms. La carte de configuration ne peut être référencée par les volumes de données que si elle se trouve dans le même espace de noms.

```
$ oc get ns
```

2. Créer la carte de configuration :

```
oc create configmap <configmap-name> --from-file=</path/to/file/ca.pem>
```

10.19.1.2. Exemple : Carte de configuration créée à partir d'un certificat TLS

L'exemple suivant est celui d'une carte de configuration créée à partir du certificat TLS de **ca.pem**.

-

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: tls-certs
data:
  ca.pem: |
    -----BEGIN CERTIFICATE-----
    ... <base64 encoded cert> ...
    -----END CERTIFICATE-----
```

10.19.2. Importation d'images de machines virtuelles avec des volumes de données

Utilisez l'importateur de données conteneurisées (CDI) pour importer une image de machine virtuelle dans une réclamation de volume persistant (PVC) à l'aide d'un volume de données. Vous pouvez attacher un volume de données à une machine virtuelle pour un stockage persistant.

L'image de la machine virtuelle peut être hébergée à un point de terminaison HTTP ou HTTPS, ou intégrée dans un disque de conteneur et stockée dans un registre de conteneur.

IMPORTANT

Lorsque vous importez une image de disque dans un PVC, l'image de disque est étendue pour utiliser toute la capacité de stockage demandée dans le PVC. Pour utiliser cet espace, les partitions du disque et le(s) système(s) de fichiers de la machine virtuelle peuvent avoir besoin d'être étendus.

La procédure de redimensionnement varie en fonction du système d'exploitation installé sur la machine virtuelle. Consultez la documentation du système d'exploitation pour plus de détails.

10.19.2.1. Conditions préalables

- Si le point de terminaison nécessite un certificat TLS, le certificat doit être inclus dans une carte de configuration dans le même espace de noms que le volume de données et référencé dans la configuration du volume de données.
- Pour importer un disque de conteneur :
  - Il se peut que vous deviez préparer un disque de conteneur à partir d'une image de machine virtuelle et le stocker dans votre registre de conteneur avant de l'importer.
  - Si le registre de conteneurs ne dispose pas de TLS, vous devez ajouter le registre au champ **insecureRegistries** de la ressource personnalisée **HyperConverged** avant de pouvoir importer un disque de conteneur à partir de ce registre.
- Il se peut que vous deviez définir une classe de stockage ou préparer l'espace scratch CDI pour que cette opération se déroule correctement.

10.19.2.2. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

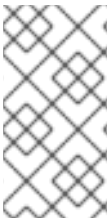
** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

NOTE

CDI utilise désormais la configuration du proxy à l'échelle du cluster de OpenShift Container Platform.

10.19.2.3. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.19.2.4. Importation d'une image de machine virtuelle dans le stockage à l'aide d'un volume de données

Vous pouvez importer une image de machine virtuelle dans le stockage en utilisant un volume de données.

L'image de la machine virtuelle peut être hébergée à un point de terminaison HTTP ou HTTPS ou l'image peut être intégrée dans un disque de conteneur et stockée dans un registre de conteneur.

Vous spécifiez la source de données pour l'image dans un fichier de configuration **VirtualMachine**. Lorsque la machine virtuelle est créée, le volume de données avec l'image de la machine virtuelle est importé dans le stockage.

Conditions préalables

- Pour importer une image de machine virtuelle, vous devez disposer des éléments suivants :
  - Une image de disque de machine virtuelle au format RAW, ISO ou QCOW2, éventuellement compressée à l'aide de **xz** ou **gz**.
  - Un point d'accès HTTP ou HTTPS où l'image est hébergée, ainsi que les informations d'authentification nécessaires pour accéder à la source de données.
- Pour importer un disque conteneur, vous devez disposer d'une image de machine virtuelle intégrée dans un disque conteneur et stockée dans un registre de conteneur, ainsi que des informations d'authentification nécessaires pour accéder à la source de données.
- Si la machine virtuelle doit communiquer avec des serveurs qui utilisent des certificats auto-signés ou des certificats non signés par le groupe d'autorités de certification du système, vous devez créer une carte de configuration dans le même espace de noms que le volume de données.

Procédure

1. Si votre source de données nécessite une authentification, créez un manifeste **Secret**, en spécifiant les informations d'identification de la source de données, et enregistrez-le sous **endpoint-secret.yaml**:

```
apiVersion: v1
kind: Secret
metadata:
  name: endpoint-secret 1
  labels:
    app: containerized-data-importer
type: Opaque
data:
  accessKeyId: "" 2
  secretKey:   "" 3
```

1. Indiquez le nom de l'adresse **Secret**.
2. Indiquez l'ID de la clé ou le nom de l'utilisateur codé en Base64.
3. Spécifiez la clé secrète ou le mot de passe codé en Base64.

2. Appliquer le manifeste **Secret**:

```
$ oc apply -f endpoint-secret.yaml
```

3. Modifiez le manifeste **VirtualMachine**, en spécifiant la source de données pour l'image de machine virtuelle que vous souhaitez importer, et enregistrez-le sous **vm-fedora-datavolume.yaml**:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  creationTimestamp: null
  labels:
    kubevirt.io/vm: vm-fedora-datavolume
  name: vm-fedora-datavolume (1)
spec:
  dataVolumeTemplates:
  - metadata:
      creationTimestamp: null
      name: fedora-dv (2)
    spec:
      storage:
        resources:
          requests:
            storage: 10Gi
        storageClassName: local
      source:
        http: (3)
          url: "https://mirror.arizona.edu/fedora/linux/releases/35/Cloud/x86_64/images/Fedora-Cloud-Base-35-1.2.x86_64.qcow2" (4)
          secretRef: endpoint-secret (5)
          certConfigMap: "" (6)
    status: {}
  running: true
  template:
    metadata:
      creationTimestamp: null
      labels:
        kubevirt.io/vm: vm-fedora-datavolume
    spec:
      domain:
        devices:
          disks:
          - disk:
              bus: virtio
            name: datavolumedisk1
        machine:
          type: ""
        resources:
          requests:
            memory: 1.5Gi
      terminationGracePeriodSeconds: 180
      volumes:
      - dataVolume:
          name: fedora-dv
        name: datavolumedisk1
status: {}
```

(1) Indiquez le nom de la machine virtuelle.

2 Indiquez le nom du volume de données.

3 Indiquez **http** pour un point de terminaison HTTP ou HTTPS. Indiquez **registry** pour une image de disque de conteneur importée d'un registre.

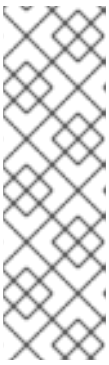
4 Spécifiez l'URL ou le point de terminaison du registre de l'image de machine virtuelle que vous souhaitez importer. Cet exemple référence une image de machine virtuelle à un point de terminaison HTTPS. Un exemple de point d'extrémité de registre de conteneur est **url: "docker://kubevirt/fedora-cloud-container-disk-demo:latest"**.

5 Indiquez le nom **Secret** si vous avez créé un **Secret** pour la source de données.

6 En option : Spécifiez une carte de configuration de certificat CA.

4. Créer la machine virtuelle :

```
$ oc create -f vm-fedora-datavolume.yaml
```



NOTE

La commande **oc create** crée le volume de données et la machine virtuelle. Le contrôleur CDI crée un PVC sous-jacent avec l'annotation correcte et le processus d'importation commence. Lorsque l'importation est terminée, l'état du volume de données devient **Succeeded**. Vous pouvez démarrer la machine virtuelle.

L'approvisionnement en volume de données s'effectue en arrière-plan, il n'est donc pas nécessaire de surveiller le processus.

Vérification

1. Le pod importateur télécharge l'image de la machine virtuelle ou le disque du conteneur à partir de l'URL spécifiée et le stocke sur le PV provisionné. Affichez l'état du module d'importation en exécutant la commande suivante :

```
$ oc get pods
```

2. Surveillez le volume de données jusqu'à ce que son état soit **Succeeded** en exécutant la commande suivante :

```
oc describe dv fedora-dv 1
```

1 Indiquez le nom du volume de données que vous avez défini dans le manifeste **VirtualMachine**.

3. Vérifiez que le provisionnement est terminé et que la machine virtuelle a démarré en accédant à sa console série :

```
$ virtctl console vm-fedora-datavolume
```

10.19.2.5. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.19.3. Importation d'images de machines virtuelles dans le stockage en bloc avec des volumes de données

Vous pouvez importer une image de machine virtuelle existante dans votre cluster OpenShift Container Platform. OpenShift Virtualization utilise des volumes de données pour automatiser l'importation de données et la création d'une revendication de volume persistant (PVC) sous-jacente.

IMPORTANT

Lorsque vous importez une image de disque dans un PVC, l'image de disque est étendue pour utiliser toute la capacité de stockage demandée dans le PVC. Pour utiliser cet espace, les partitions du disque et le(s) système(s) de fichiers de la machine virtuelle peuvent avoir besoin d'être étendus.

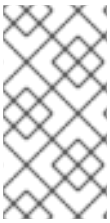
La procédure de redimensionnement varie en fonction du système d'exploitation installé sur la machine virtuelle. Consultez la documentation du système d'exploitation pour plus de détails.

10.19.3.1. Conditions préalables

- Si vous avez besoin d'un espace scratch conformément à la matrice des opérations supportées par le CDI, vous devez d'abord définir une classe de stockage ou préparer l'espace scratch du CDI pour que cette opération se déroule correctement.

10.19.3.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.19.3.3. À propos des volumes persistants en bloc

Un volume persistant (PV) en mode bloc est un PV soutenu par un périphérique en mode bloc brut. Ces volumes n'ont pas de système de fichiers et peuvent offrir des avantages en termes de performances pour les machines virtuelles en réduisant les frais généraux.

Les volumes de blocs bruts sont approvisionnés en spécifiant **volumeMode: Block** dans les spécifications PV et PVC (persistent volume claim).

10.19.3.4. Création d'un volume persistant en bloc local

Créez un volume persistant (PV) local en bloc sur un nœud en remplissant un fichier et en le montant en tant que périphérique en boucle. Vous pouvez ensuite référencer ce périphérique en boucle dans un manifeste PV en tant que volume **Block** et l'utiliser comme périphérique de bloc pour une image de machine virtuelle.

Procédure

1. Connectez-vous en tant que **root** au nœud sur lequel vous souhaitez créer le PV local. Cette procédure utilise **node01** pour ses exemples.

2. Créez un fichier et remplissez-le de caractères nuls afin qu'il puisse être utilisé comme périphérique de bloc. L'exemple suivant crée un fichier **loop10** d'une taille de 2 Go (20 blocs de 100 Mo) :

```
$ dd if=/dev/zero of=<loop10> bs=100M count=20
```

3. Monter le fichier **loop10** en tant que périphérique en boucle.

```
$ losetup </dev/loop10>d3 <loop10> 1 2
```

1. Chemin d'accès au fichier où le périphérique loop est monté.
2. Le fichier créé à l'étape précédente doit être monté en tant que périphérique de boucle.

4. Créez un manifeste **PersistentVolume** qui fait référence au périphérique en boucle monté.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: <local-block-pv10>
  annotations:
spec:
  local:
    path: </dev/loop10> 1
  capacity:
    storage: <2Gi>
  volumeMode: Block 2
  storageClassName: local 3
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - <node01> 4
```

(1) Chemin d'accès du dispositif de boucle sur le nœud.

(2) Indique qu'il s'agit d'un PV en bloc.

(3) Facultatif : Définissez une classe de stockage pour le PV. Si vous ne le faites pas, la valeur par défaut du cluster est utilisée.

(4) Le nœud sur lequel le périphérique de bloc a été monté.

5. Créer le bloc PV.

```
# oc create -f <local-block-pv10.yaml>1
```

(1) Le nom de fichier du volume persistant créé à l'étape précédente.

10.19.3.5. Importation d'une image de machine virtuelle dans le stockage en bloc à l'aide d'un volume de données

Vous pouvez importer une image de machine virtuelle dans le stockage en bloc à l'aide d'un volume de données. Vous faites référence au volume de données dans un manifeste **VirtualMachine** avant de créer une machine virtuelle.

Conditions préalables

- Une image de disque de machine virtuelle au format RAW, ISO ou QCOW2, éventuellement compressée à l'aide de **xz** ou **gz**.
- Un point d'accès HTTP ou HTTPS où l'image est hébergée, ainsi que les informations d'authentification nécessaires pour accéder à la source de données.

Procédure

1. Si votre source de données nécessite une authentification, créez un manifeste **Secret**, en spécifiant les informations d'identification de la source de données, et enregistrez-le sous **endpoint-secret.yaml**:

```
apiVersion: v1
kind: Secret
metadata:
  name: endpoint-secret 1
  labels:
    app: containerized-data-importer
type: Opaque
data:
  accessKeyId: "" 2
  secretKey:   "" 3
```

(1) Indiquez le nom de l'adresse **Secret**.

(2) Indiquez l'ID de la clé ou le nom de l'utilisateur codé en Base64.

(3) Spécifiez la clé secrète ou le mot de passe codé en Base64.

2. Appliquer le manifeste **Secret**:

```
$ oc apply -f endpoint-secret.yaml
```

3. Créez un manifeste **DataVolume**, en spécifiant la source de données pour l'image de la machine virtuelle et **Block** pour **storage.volumeMode**.

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: import-pv-datavolume 1
spec:
  storageClassName: local 2
    source:
      http:
        url: "https://mirror.arizona.edu/fedora/linux/releases/35/Cloud/x86_64/images/Fedora-Cloud-Base-35-1.2.x86_64.qcow2" 3
        secretRef: endpoint-secret 4
  storage:
    volumeMode: Block 5
    resources:
      requests:
        storage: 10Gi
```

1. Indiquez le nom du volume de données.
2. Facultatif : Définissez la classe de stockage ou omettez-la pour accepter la valeur par défaut du cluster.
3. Spécifiez l'URL HTTP ou HTTPS de l'image à importer.
4. Indiquez le nom **Secret** si vous avez créé un **Secret** pour la source de données.
5. Le mode de volume et le mode d'accès sont détectés automatiquement pour les fournisseurs de stockage connus. Dans le cas contraire, indiquez **Block**.

4. Créez le volume de données pour importer l'image de la machine virtuelle :

```
$ oc create -f import-pv-datavolume.yaml
```

Vous pouvez faire référence à ce volume de données dans un manifeste **VirtualMachine** avant de créer une machine virtuelle.

10.19.3.6. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

NOTE

CDI utilise désormais la configuration du proxy à l'échelle du cluster de OpenShift Container Platform.

10.19.3.7. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.20. CLONAGE DE MACHINES VIRTUELLES

10.20.1. Permettre aux utilisateurs de cloner des volumes de données entre espaces de noms

La nature isolante des espaces de noms signifie que les utilisateurs ne peuvent pas, par défaut, cloner les ressources entre les espaces de noms.

Pour permettre à un utilisateur de cloner une machine virtuelle dans un autre espace de noms, un utilisateur ayant le rôle **cluster-admin** doit créer un nouveau rôle de cluster. Liez ce rôle de cluster à un utilisateur pour lui permettre de cloner des machines virtuelles dans l'espace de noms de destination.

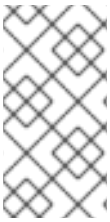
10.20.1.1. Conditions préalables

- Seul un utilisateur ayant le rôle **cluster-admin** peut créer des rôles de cluster.

10.20.1.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous

pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.20.1.3. Création de ressources RBAC pour le clonage de volumes de données

Créez un nouveau rôle de cluster qui active les permissions pour toutes les actions de la ressource **datavolumes**.

Procédure

1. Créer un manifeste **ClusterRole**:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: <datavolume-cloner> 1
rules:
- apiGroups: ["cdi.kubevirt.io"]
  resources: ["datavolumes/source"]
  verbs: ["*"]
```

1 Nom unique pour le rôle de cluster.

2. Créer le rôle de cluster dans le cluster :

```
oc create -f <datavolume-cloner.yaml> 1
```

1 Le nom du fichier du manifeste **ClusterRole** créé à l'étape précédente.

3. Créez un manifeste **RoleBinding** qui s'applique aux espaces de noms source et destination et qui fait référence au rôle de cluster créé à l'étape précédente.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: <allow-clone-to-user> 1
  namespace: <Source namespace> 2
subjects:
- kind: ServiceAccount
  name: default
```

```
  namespace: <Destination namespace> 3
roleRef:
  kind: ClusterRole
  name: datavolume-cloner 4
  apiGroup: rbac.authorization.k8s.io
```

1. Nom unique pour la liaison de rôle.
2. L'espace de noms du volume de données source.
3. L'espace de noms dans lequel le volume de données est cloné.
4. Le nom du rôle de cluster créé à l'étape précédente.

4. Créez la liaison de rôle dans le cluster :

```
oc create -f <datavolume-cloner.yaml> 1
```

1. Le nom du fichier du manifeste **RoleBinding** créé à l'étape précédente.

10.20.2. Clonage d'un disque de machine virtuelle dans un nouveau volume de données

Vous pouvez cloner la revendication de volume persistant (PVC) d'un disque de machine virtuelle dans un nouveau volume de données en faisant référence au PVC source dans votre fichier de configuration du volume de données.

AVERTISSEMENT

Les opérations de clonage entre différents modes de volume sont prises en charge, comme le clonage d'un volume persistant (PV) avec **volumeMode: Block** vers un PV avec **volumeMode: Filesystem**.

Cependant, vous ne pouvez cloner entre différents modes de volume que s'ils sont du même type que **contentType: kubevirt**.

ASTUCE

Lorsque vous activez la pré-allocation globalement ou pour un seul volume de données, l'importateur de données conteneurisées (CDI) pré-allocation l'espace disque pendant le clonage. La pré-allocation améliore les performances d'écriture. Pour plus d'informations, voir Utilisation de la pré-allocation pour les volumes de données.

10.20.2.1. Conditions préalables

- Les utilisateurs ont besoin d'autorisations supplémentaires pour cloner le PVC d'un disque de machine virtuelle dans un autre espace de noms.

10.20.2.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.20.2.3. Clonage de la revendication de volume persistant d'un disque de machine virtuelle dans un nouveau volume de données

Vous pouvez cloner une revendication de volume persistant (PVC) d'un disque de machine virtuelle existant dans un nouveau volume de données. Le nouveau volume de données peut alors être utilisé pour une nouvelle machine virtuelle.

NOTE

Lorsqu'un volume de données est créé indépendamment d'une machine virtuelle, le cycle de vie du volume de données est indépendant de la machine virtuelle. Si la machine virtuelle est supprimée, ni le volume de données ni son PVC associé ne sont supprimés.

Conditions préalables

- Déterminez le PVC d'un disque de machine virtuelle existant à utiliser. Vous devez mettre hors tension la machine virtuelle associée au PVC avant de pouvoir la cloner.
- Installez le CLI OpenShift (**oc**).

Procédure

1. Examinez le disque de la machine virtuelle que vous souhaitez cloner pour identifier le nom et l'espace de noms du PVC associé.
2. Créer un fichier YAML pour un volume de données qui spécifie le nom du nouveau volume de données, le nom et l'espace de noms du PVC source et la taille du nouveau volume de données. Par exemple :

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <cloner-datavolume> 1
spec:
```

```
source:
  pvc:
    namespace: "<source-namespace>" 2
    name: "<my-favorite-vm-disk>" 3
pvc:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: <2Gi> 4
```

1. Le nom du nouveau volume de données.
2. L'espace de noms dans lequel le PVC source existe.
3. Le nom du PVC source.
4. La taille du nouveau volume de données. Vous devez allouer suffisamment d'espace, sinon l'opération de clonage échoue. La taille doit être égale ou supérieure à celle du PVC source.

3. Commencez à cloner le PVC en créant le volume de données :

```
oc create -f <cloner-datavolume>.yaml
```

NOTE

Les volumes de données empêchent le démarrage d'une machine virtuelle avant que le PVC ne soit préparé. Vous pouvez donc créer une machine virtuelle qui fait référence au nouveau volume de données pendant que le PVC se clone.

10.20.2.4. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.20.3. Clonage d'une machine virtuelle à l'aide d'un modèle de volume de données

Vous pouvez créer une nouvelle machine virtuelle en clonant la revendication de volume persistant (PVC) d'une VM existante. En incluant une adresse **dataVolumeTemplate** dans le fichier de configuration de votre machine virtuelle, vous créez un nouveau volume de données à partir du PVC d'origine.

AVERTISSEMENT

Les opérations de clonage entre différents modes de volume sont prises en charge, comme le clonage d'un volume persistant (PV) avec **volumeMode: Block** vers un PV avec **volumeMode: Filesystem**.

Cependant, vous ne pouvez cloner entre différents modes de volume que s'ils sont du même type que **contentType: kubevirt**.

ASTUCE

Lorsque vous activez la pré-allocation globalement ou pour un seul volume de données, l'importateur de données conteneurisées (CDI) pré-allocation l'espace disque pendant le clonage. La pré-allocation améliore les performances d'écriture. Pour plus d'informations, voir Utilisation de la pré-allocation pour les volumes de données.

10.20.3.1. Conditions préalables

- Les utilisateurs ont besoin d'autorisations supplémentaires pour cloner le PVC d'un disque de machine virtuelle dans un autre espace de noms.

10.20.3.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de

données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.20.3.3. Création d'une nouvelle machine virtuelle à partir d'une revendication de volume persistant cloné à l'aide d'un modèle de volume de données

Vous pouvez créer une machine virtuelle qui clone la revendication de volume persistant (PVC) d'une machine virtuelle existante dans un volume de données. Faites référence à **dataVolumeTemplate** dans le manifeste de la machine virtuelle et le PVC **source** est cloné dans un volume de données, qui est ensuite automatiquement utilisé pour la création de la machine virtuelle.

NOTE

Lorsqu'un volume de données est créé dans le cadre du modèle de volume de données d'une machine virtuelle, le cycle de vie du volume de données dépend alors de la machine virtuelle. Si la machine virtuelle est supprimée, le volume de données et le PVC associé sont également supprimés.

Conditions préalables

- Déterminez le PVC d'un disque de machine virtuelle existant à utiliser. Vous devez mettre hors tension la machine virtuelle associée au PVC avant de pouvoir la cloner.
- Installez le CLI OpenShift (**oc**).

Procédure

1. Examinez la machine virtuelle que vous souhaitez cloner pour identifier le nom et l'espace de noms du PVC associé.
2. Créez un fichier YAML pour un objet **VirtualMachine**. L'exemple de machine virtuelle suivant clone **my-favorite-vm-disk**, qui est situé dans l'espace de noms **source-namespace**. Le volume de données **2Gi** appelé **favorite-clone** est créé à partir de **my-favorite-vm-disk**.
   Par exemple :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    kubevirt.io/vm: vm-dv-clone
  name: vm-dv-clone (1)
spec:
  running: false
  template:
    metadata:
      labels:
        kubevirt.io/vm: vm-dv-clone
    spec:
      domain:
        devices:
          disks:
          - disk:
              bus: virtio
            name: root-disk
        resources:
```

```
        requests:
          memory: 64M
    volumes:
    - dataVolume:
        name: favorite-clone
      name: root-disk
dataVolumeTemplates:
- metadata:
    name: favorite-clone
  spec:
    storage:
      accessModes:
      - ReadWriteOnce
      resources:
        requests:
          storage: 2Gi
    source:
      pvc:
        namespace: "source-namespace"
        name: "my-favorite-vm-disk"
```

1 La machine virtuelle à créer.

3. Créez la machine virtuelle avec le volume de données cloné par le PVC :

```
$ oc create -f <vm-clone-datavolumetemplate>.yaml
```

10.20.3.4. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.20.4. Clonage d'un disque de machine virtuelle dans un nouveau volume de données de stockage par blocs

Vous pouvez cloner la revendication de volume persistant (PVC) d'un disque de machine virtuelle dans un nouveau volume de données en bloc en référençant le PVC source dans votre fichier de configuration du volume de données.

AVERTISSEMENT

Les opérations de clonage entre différents modes de volume sont prises en charge, comme le clonage d'un volume persistant (PV) avec **volumeMode: Block** vers un PV avec **volumeMode: Filesystem**.

Cependant, vous ne pouvez cloner entre différents modes de volume que s'ils sont du même type que **contentType: kubevirt**.

ASTUCE

Lorsque vous activez la pré-allocation globalement ou pour un seul volume de données, l'importateur de données conteneurisées (CDI) pré-allocation l'espace disque pendant le clonage. La pré-allocation améliore les performances d'écriture. Pour plus d'informations, voir Utilisation de la pré-allocation pour les volumes de données.

10.20.4.1. Conditions préalables

- Les utilisateurs ont besoin d'autorisations supplémentaires pour cloner le PVC d'un disque de machine virtuelle dans un autre espace de noms.

10.20.4.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.20.4.3. À propos des volumes persistants en bloc

Un volume persistant (PV) en mode bloc est un PV soutenu par un périphérique en mode bloc brut. Ces volumes n'ont pas de système de fichiers et peuvent offrir des avantages en termes de performances pour les machines virtuelles en réduisant les frais généraux.

Les volumes de blocs bruts sont approvisionnés en spécifiant **volumeMode: Block** dans les spécifications PV et PVC (persistent volume claim).

10.20.4.4. Création d'un volume persistant en bloc local

Créez un volume persistant (PV) local en bloc sur un nœud en remplissant un fichier et en le montant en tant que périphérique en boucle. Vous pouvez ensuite référencer ce périphérique en boucle dans un manifeste PV en tant que volume **Block** et l'utiliser comme périphérique de bloc pour une image de machine virtuelle.

Procédure

1. Connectez-vous en tant que **root** au nœud sur lequel vous souhaitez créer le PV local. Cette procédure utilise **node01** pour ses exemples.

2. Créez un fichier et remplissez-le de caractères nuls afin qu'il puisse être utilisé comme périphérique de bloc. L'exemple suivant crée un fichier **loop10** d'une taille de 2 Go (20 blocs de 100 Mo) :

```
$ dd if=/dev/zero of=<loop10> bs=100M count=20
```

3. Monter le fichier **loop10** en tant que périphérique en boucle.

```
$ losetup </dev/loop10>d3 <loop10> 1 2
```

1. Chemin d'accès au fichier où le périphérique loop est monté.
2. Le fichier créé à l'étape précédente doit être monté en tant que périphérique de boucle.

4. Créez un manifeste **PersistentVolume** qui fait référence au périphérique en boucle monté.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: <local-block-pv10>
  annotations:
spec:
  local:
    path: </dev/loop10> 1
  capacity:
    storage: <2Gi>
  volumeMode: Block 2
  storageClassName: local 3
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
```

```
      - key: kubernetes.io/hostname
        operator: In
        values:
        - <node01> 4
```

1. Chemin d'accès du dispositif de boucle sur le nœud.
2. Indique qu'il s'agit d'un PV en bloc.
3. Facultatif : Définissez une classe de stockage pour le PV. Si vous ne le faites pas, la valeur par défaut du cluster est utilisée.
4. Le nœud sur lequel le périphérique de bloc a été monté.

5. Créer le bloc PV.

```
# oc create -f <local-block-pv10.yaml> 1
```

1. Le nom de fichier du volume persistant créé à l'étape précédente.

10.20.4.5. Clonage de la revendication de volume persistant d'un disque de machine virtuelle dans un nouveau volume de données

Vous pouvez cloner une revendication de volume persistant (PVC) d'un disque de machine virtuelle existant dans un nouveau volume de données. Le nouveau volume de données peut alors être utilisé pour une nouvelle machine virtuelle.

NOTE

Lorsqu'un volume de données est créé indépendamment d'une machine virtuelle, le cycle de vie du volume de données est indépendant de la machine virtuelle. Si la machine virtuelle est supprimée, ni le volume de données ni son PVC associé ne sont supprimés.

Conditions préalables

- Déterminez le PVC d'un disque de machine virtuelle existant à utiliser. Vous devez mettre hors tension la machine virtuelle associée au PVC avant de pouvoir la cloner.
- Installez le CLI OpenShift (**oc**).
- Au moins un volume persistant (PV) en bloc disponible de taille identique ou supérieure à celle du PVC source.

Procédure

1. Examinez le disque de la machine virtuelle que vous souhaitez cloner pour identifier le nom et l'espace de noms du PVC associé.
2. Créer un fichier YAML pour un volume de données qui spécifie le nom du nouveau volume de données, le nom et l'espace de noms du PVC source, **volumeMode: Block** pour qu'un PV de bloc disponible soit utilisé, et la taille du nouveau volume de données.
   Par exemple :

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <cloner-datavolume> 1
spec:
  source:
    pvc:
      namespace: "<source-namespace>" 2
      name: "<my-favorite-vm-disk>" 3
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: <2Gi> 4
    volumeMode: Block 5
```

1. Le nom du nouveau volume de données.
2. L'espace de noms dans lequel le PVC source existe.
3. Le nom du PVC source.
4. La taille du nouveau volume de données. Vous devez allouer suffisamment d'espace, sinon l'opération de clonage échoue. La taille doit être égale ou supérieure à celle du PVC source.
5. Spécifie que la destination est un bloc PV

3. Commencez à cloner le PVC en créant le volume de données :

```
oc create -f <cloner-datavolume>.yaml
```

NOTE

Les volumes de données empêchent le démarrage d'une machine virtuelle avant que le PVC ne soit préparé. Vous pouvez donc créer une machine virtuelle qui fait référence au nouveau volume de données pendant que le PVC se clone.

10.20.4.6. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.21. MISE EN RÉSEAU DE MACHINES VIRTUELLES

10.21.1. Configuration de la machine virtuelle pour le réseau de pods par défaut

Vous pouvez connecter une machine virtuelle au réseau de pods interne par défaut en configurant son interface réseau pour qu'elle utilise le mode de liaison **masquerade**

10.21.1.1. Configuration du mode mascarade à partir de la ligne de commande

Vous pouvez utiliser le mode mascarade pour cacher le trafic sortant d'une machine virtuelle derrière l'adresse IP du pod. Le mode mascarade utilise la traduction d'adresses réseau (NAT) pour connecter les machines virtuelles au backend du réseau du pod par l'intermédiaire d'un pont Linux.

Activez le mode mascarade et autorisez le trafic à entrer dans la machine virtuelle en modifiant le fichier de configuration de votre machine virtuelle.

Conditions préalables

- La machine virtuelle doit être configurée pour utiliser DHCP afin d'acquérir des adresses IPv4. Les exemples ci-dessous sont configurés pour utiliser DHCP.

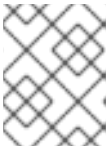
Procédure

1. Modifiez la spécification **interfaces** du fichier de configuration de votre machine virtuelle :

```
kind: VirtualMachine
spec:
  domain:
    devices:
      interfaces:
        - name: default
          masquerade: {} 1
          ports: 2
            - port: 80
```

```
networks:
- name: default
  pod: {}
```

1. Se connecter en utilisant le mode mascarade.
2. Facultatif : Listez les ports que vous souhaitez exposer à partir de la machine virtuelle, chacun étant spécifié par le champ **port**. La valeur **port** doit être un nombre compris entre 0 et 65536. Lorsque le tableau **ports** n'est pas utilisé, tous les ports de la plage valide sont ouverts au trafic entrant. Dans cet exemple, le trafic entrant est autorisé sur le port **80**.



NOTE

Les ports 49152 et 49153 sont réservés à l'usage de la plateforme libvirt et tout autre trafic entrant vers ces ports est supprimé.

2. Créer la machine virtuelle :

```
$ oc create -f <vm-name>.yaml
```

10.21.1.2. Configuration du mode masquerade avec dual-stack (IPv4 et IPv6)

Vous pouvez configurer une nouvelle machine virtuelle (VM) pour qu'elle utilise à la fois IPv6 et IPv4 sur le réseau de pods par défaut à l'aide de cloud-init.

Le champ **Network.pod.vmIPv6NetworkCIDR** dans la configuration de l'instance de la machine virtuelle détermine l'adresse IPv6 statique de la VM et l'adresse IP de la passerelle. Ces adresses sont utilisées par le pod virt-launcher pour acheminer le trafic IPv6 vers la machine virtuelle et ne sont pas utilisées à l'extérieur. Le champ **Network.pod.vmIPv6NetworkCIDR** spécifie un bloc d'adresses IPv6 en notation CIDR (Classless Inter-Domain Routing). La valeur par défaut est **fd10:0:2::2/120**. Vous pouvez modifier cette valeur en fonction des exigences de votre réseau.

Lorsque la machine virtuelle fonctionne, le trafic entrant et sortant de la machine virtuelle est acheminé vers l'adresse IPv4 et l'adresse IPv6 unique du module de lancement virtuel. Le module virt-launcher achemine ensuite le trafic IPv4 vers l'adresse DHCP de la machine virtuelle et le trafic IPv6 vers l'adresse IPv6 statique de la machine virtuelle.

Conditions préalables

- Le cluster OpenShift Container Platform doit utiliser le plugin réseau OVN-Kubernetes Container Network Interface (CNI) configuré pour la double pile.

Procédure

1. Dans une nouvelle configuration de machine virtuelle, incluez une interface avec **masquerade** et configurez l'adresse IPv6 et la passerelle par défaut à l'aide de cloud-init.

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: example-vm-ipv6
...
        interfaces:
```

```
        - name: default
          masquerade: {} (1)
          ports:
            - port: 80 (2)
      networks:
      - name: default
        pod: {}
      volumes:
      - cloudInitNoCloud:
          networkData: |
            version: 2
            ethernets:
              eth0:
                dhcp4: true
                addresses: [ fd10:0:2::2/120 ] (3)
                gateway6: fd10:0:2::1 (4)
```

1. Se connecter en utilisant le mode mascarade.
2. Autorise le trafic entrant sur le port 80 vers la machine virtuelle.
3. L'adresse IPv6 statique déterminée par le champ **Network.pod.vmIPv6NetworkCIDR** dans la configuration de l'instance de la machine virtuelle. La valeur par défaut est **fd10:0:2::2/120**.
4. L'adresse IP de la passerelle déterminée par le champ **Network.pod.vmIPv6NetworkCIDR** dans la configuration de l'instance de la machine virtuelle. La valeur par défaut est **fd10:0:2::1**.

2. Créer la machine virtuelle dans l'espace de noms :

```
$ oc create -f example-vm-ipv6.yaml
```

Vérification

- Pour vérifier qu'IPv6 a été configuré, démarrez la machine virtuelle et affichez l'état de l'interface de l'instance de la machine virtuelle pour vous assurer qu'elle dispose d'une adresse IPv6 :

```
$ oc get vmi <vmi-name> -o jsonpath="{.status.interfaces[*].ipAddresses}"
```

10.21.2. Création d'un service pour exposer une machine virtuelle

Vous pouvez exposer une machine virtuelle à l'intérieur ou à l'extérieur du cluster en utilisant un objet **Service**.

10.21.2.1. À propos des services

Un Kubernetes *service* est un moyen abstrait d'exposer une application s'exécutant sur un ensemble de pods en tant que service réseau. Les services permettent à vos applications de recevoir du trafic. Les services peuvent être exposés de différentes manières en spécifiant un **spec.type** dans l'objet **Service**:

ClusterIP

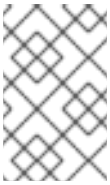
Expose le service sur une adresse IP interne au sein du cluster. **ClusterIP** est le service par défaut **type**.

NodePort

Expose le service sur le même port de chaque nœud sélectionné dans la grappe. **NodePort** rend un service accessible depuis l'extérieur de la grappe.

Équilibreur de charge

Crée un équilibreur de charge externe dans le nuage actuel (s'il est pris en charge) et attribue une adresse IP externe fixe au service.



NOTE

Pour les clusters sur site, vous pouvez configurer un service d'équilibrage de charge en utilisant l'opérateur MetalLB en mode couche 2. Le mode BGP n'est pas pris en charge. Le MetalLB Operator est installé dans l'espace de noms **metallb-system**.

Ressources supplémentaires

- Installation de l'opérateur MetalLB
- Configuration des services pour l'utilisation de MetalLB

10.21.2.1.1. Prise en charge de la double pile

Si le réseau à double pile IPv4 et IPv6 est activé pour votre cluster, vous pouvez créer un service qui utilise IPv4, IPv6 ou les deux, en définissant les champs **spec.ipFamilyPolicy** et **spec.ipFamilies** dans l'objet **Service**.

Le champ **spec.ipFamilyPolicy** peut prendre l'une des valeurs suivantes :

Pile unique

Le plan de contrôle attribue une adresse IP de cluster pour le service sur la base de la première plage IP de cluster de service configurée.

PreferDualStack

Le plan de contrôle attribue des adresses IP IPv4 et IPv6 pour le service sur les clusters configurés en double pile.

RequireDualStack

Cette option échoue pour les clusters dont le réseau à double pile n'est pas activé. Pour les clusters dont la double pile est configurée, le comportement est le même que lorsque la valeur est définie sur **PreferDualStack**. Le plan de contrôle alloue les adresses IP des clusters à partir des plages d'adresses IPv4 et IPv6.

Vous pouvez définir la famille IP à utiliser pour la pile simple ou définir l'ordre des familles IP pour la pile double en définissant le champ **spec.ipFamilies** sur l'une des valeurs de tableau suivantes :

- **[IPv4]**
- **[IPv6]**
- **[IPv4, IPv6]**
- **[IPv6, IPv4]**

10.21.2.2. Exposer une machine virtuelle en tant que service

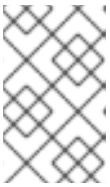
Créez un service **ClusterIP**, **NodePort** ou **LoadBalancer** pour vous connecter à une machine virtuelle (VM) en cours d'exécution depuis l'intérieur ou l'extérieur du cluster.

Procédure

1. Modifiez le manifeste **VirtualMachine** pour ajouter l'étiquette de création de service :

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: vm-ephemeral
  namespace: example-namespace
spec:
  running: false
  template:
    metadata:
      labels:
        special: key (1)
# ...
```

(1) Ajouter le libellé **special: key** dans la section **spec.template.metadata.labels**.



NOTE

Les étiquettes d'une machine virtuelle sont transmises au pod. L'étiquette **special: key** doit correspondre à l'étiquette de l'attribut **spec.selector** du manifeste **Service**.

2. Enregistrez le fichier manifeste **VirtualMachine** pour appliquer vos modifications.

3. Créez un manifeste **Service** pour exposer la VM :

```
apiVersion: v1
kind: Service
metadata:
  name: vmservice (1)
  namespace: example-namespace (2)
spec:
  externalTrafficPolicy: Cluster (3)
  ports:
  - nodePort: 30000 (4)
    port: 27017
    protocol: TCP
    targetPort: 22 (5)
  selector:
    special: key (6)
  type: NodePort (7)
```

(1) Le nom de l'objet **Service**.

(2) L'espace de noms dans lequel réside l'objet **Service**. Il doit correspondre au champ

3. Facultatif : Spécifie comment les nœuds distribuent le trafic de service reçu sur des adresses IP externes. Cette option ne s'applique qu'aux types de service **NodePort** et
4. Facultatif : Lorsqu'elle est définie, la valeur **nodePort** doit être unique pour tous les services. Si elle n'est pas spécifiée, une valeur comprise dans la plage supérieure à **30000** est attribuée dynamiquement.
5. Facultatif : Le port VM à exposer par le service. Il doit faire référence à un port ouvert si une liste de ports est définie dans le manifeste VM. Si **targetPort** n'est pas spécifié, il prend la même valeur que **port**.
6. La référence à l'étiquette que vous avez ajoutée dans la strophe **spec.template.metadata.labels** du manifeste **VirtualMachine**.
7. Le type de service. Les valeurs possibles sont **ClusterIP**, **NodePort** et **LoadBalancer**.

4. Enregistrez le fichier manifeste **Service**.
5. Créez le service en exécutant la commande suivante :

```
oc create -f <service_name>.yaml
```

6. Démarrez la VM. Si la VM est déjà en cours d'exécution, redémarrez-la.

Vérification

1. Interroger l'objet **Service** pour vérifier qu'il est disponible :

```
$ oc get service -n example-namespace
```

Exemple de sortie pour le service ClusterIP

```
NAME        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)     AGE
vmservice   ClusterIP   172.30.3.149   <none>        27017/TCP   2m
```

Exemple de sortie pour le service NodePort

```
NAME        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)           AGE
vmservice   NodePort    172.30.232.73  <none>        27017:30000/TCP   5m
```

Exemple de sortie pour le service LoadBalancer

```
NAME        TYPE           CLUSTER-IP     EXTERNAL-IP                  PORT(S)           AGE
vmservice   LoadBalancer   172.30.27.5    172.29.10.235,172.29.10.235  27017:31829/TCP
5s
```

2. Choisissez la méthode appropriée pour vous connecter à la machine virtuelle :
   - Pour un service **ClusterIP**, connectez-vous à la VM depuis le cluster en utilisant l'adresse IP et le port du service. Par exemple, vous pouvez vous connecter à l'adresse IP et au port du service :

```
$ ssh fedora@172.30.3.149 -p 27017
```

- Pour un service **NodePort**, connectez-vous à la VM en spécifiant l'adresse IP du nœud et le port du nœud en dehors du réseau du cluster. Par exemple, l'adresse IP du nœud et le port du nœud se trouvent en dehors du réseau du cluster :

```
$ ssh fedora@$NODE_IP -p 30000
```

- Pour un service **LoadBalancer**, utilisez le client **vinagre** pour vous connecter à votre machine virtuelle en utilisant l'adresse IP et le port publics. Les ports externes sont alloués dynamiquement.

10.21.2.3. Ressources supplémentaires

- Configuration du trafic entrant de la grappe à l'aide d'un NodePort
- Configuration du trafic entrant dans le cluster à l'aide d'un équilibreur de charge

10.21.3. Connexion d'une machine virtuelle à un réseau pont Linux

Par défaut, OpenShift Virtualization est installé avec un seul réseau de pods interne.

Vous devez créer une définition d'attachement au réseau (NAD) pour le pont Linux afin de vous connecter à des réseaux supplémentaires.

Pour attacher une machine virtuelle à un réseau supplémentaire :

1. Créer une politique de configuration du réseau du nœud de pont Linux.
2. Créer une définition de l'attachement au réseau Linux bridge.
3. Configurer la machine virtuelle pour qu'elle reconnaisse la définition de l'attachement réseau.

Pour plus d'informations sur la planification, les types d'interface et les autres activités de mise en réseau des nœuds, voir la section relative à la mise en réseau des nœuds .

10.21.3.1. Connexion au réseau par le biais de la définition de l'attachement au réseau

10.21.3.1.1. Création d'une politique de configuration du réseau de nœuds de pont Linux

Utilisez un fichier YAML du manifeste **NodeNetworkConfigurationPolicy** pour créer le pont Linux.

Conditions préalables

- Vous avez installé l'opérateur NMState de Kubernetes.

Procédure

- Créez le manifeste **NodeNetworkConfigurationPolicy**. Cet exemple contient des valeurs types que vous devez remplacer par vos propres informations.

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
```

```
  name: br1-eth1-policy (1)
spec:
  desiredState:
    interfaces:
      - name: br1 (2)
        description: Linux bridge with eth1 as a port (3)
        type: linux-bridge (4)
        state: up (5)
        ipv4:
          enabled: false (6)
        bridge:
          options:
            stp:
              enabled: false (7)
          port:
            - name: eth1 (8)
```

1. Nom de la politique.
2. Name of the interface.
3. Facultatif : description lisible par l'homme de l'interface.
4. Le type d'interface. Cet exemple crée un pont.
5. L'état demandé pour l'interface après sa création.
6. Désactive IPv4 dans cet exemple.
7. Désactive le protocole STP dans cet exemple.
8. Le NIC du nœud auquel le pont est attaché.

10.21.3.2. Création d'une définition d'attachement à un réseau de ponts Linux

AVERTISSEMENT

La configuration de la gestion des adresses IP (IPAM) dans une définition d'attachement réseau pour les machines virtuelles n'est pas prise en charge.

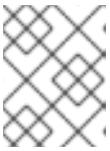
10.21.3.2.1. Création d'une définition d'attachement réseau pour un pont Linux dans la console web

Les administrateurs réseau peuvent créer des définitions d'attachement réseau pour fournir un réseau de couche 2 aux pods et aux machines virtuelles.

Procédure

1. Dans la console web, cliquez sur **Networking → Network Attachment Definitions**.

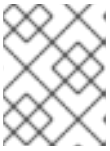
2. Cliquez sur **Create Network Attachment Definition**



NOTE

La définition de l'attachement réseau doit se trouver dans le même espace de noms que le pod ou la machine virtuelle.

3. Saisissez une adresse unique **Name** et une adresse facultative **Description**.
4. Cliquez sur la liste **Network Type** et sélectionnez **CNV Linux bridge**.
5. Entrez le nom du pont dans le champ **Bridge Name**.
6. Facultatif : si la ressource a des ID VLAN configurés, saisissez les numéros d'ID dans le champ **VLAN Tag Number**.
7. Facultatif : Sélectionnez **MAC Spoof Check** pour activer le filtrage de l'usurpation d'adresse MAC. Cette fonction offre une sécurité contre les attaques de MAC spoofing en n'autorisant qu'une seule adresse MAC à quitter le pod.
8. Cliquez sur **Create**.



NOTE

La définition de l'attachement au réseau d'un pont Linux est la méthode la plus efficace pour connecter une machine virtuelle à un VLAN.

10.21.3.2.2. Création d'une définition d'attachement à un réseau de pont Linux dans l'interface de programmation (CLI)

En tant qu'administrateur réseau, vous pouvez configurer une définition d'attachement réseau de type **cnv-bridge** pour fournir un réseau de couche 2 aux pods et aux machines virtuelles.

Conditions préalables

- Le nœud doit supporter nftables et le binaire **nft** doit être déployé pour permettre la vérification de l'usurpation de MAC.

Procédure

1. Créez une définition de pièce jointe au réseau dans le même espace de noms que la machine virtuelle.
2. Ajoutez la machine virtuelle à la définition de l'attachement réseau, comme dans l'exemple suivant :

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: <bridge-network> 1
  annotations:
    k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/<bridge-interface> 2
spec:
  config: '{
```

```
  "cniVersion": "0.3.1",
  "name": "<bridge-network>", 3
  "type": "cnv-bridge", 4
  "bridge": "<bridge-interface>", 5
  "macspoofchk": true, 6
  "vlan": 1 7
}'
```

1. Nom de l'objet **NetworkAttachmentDefinition**.
2. Facultatif : Paire clé-valeur d'annotation pour la sélection des nœuds, où **bridge-interface** doit correspondre au nom d'un pont configuré sur certains nœuds. Si vous ajoutez cette annotation à votre définition d'attachement réseau, vos instances de machines virtuelles ne s'exécuteront que sur les nœuds auxquels le pont **bridge-interface** est connecté.
3. Le nom de la configuration. Il est recommandé de faire correspondre le nom de la configuration à la valeur **name** de la définition de l'attachement réseau.
4. Le nom réel du plugin Container Network Interface (CNI) qui fournit le réseau pour cette définition d'attachement réseau. Ne modifiez pas ce champ, sauf si vous souhaitez utiliser une CNI différente.
5. Le nom du pont Linux configuré sur le nœud.
6. Facultatif : Indicateur permettant d'activer la vérification de l'usurpation d'adresse MAC. Lorsqu'il est défini sur **true**, vous ne pouvez pas modifier l'adresse MAC du pod ou de l'interface invité. Cet attribut offre une sécurité contre les attaques par usurpation d'adresse MAC en n'autorisant qu'une seule adresse MAC à quitter le pod.
7. Facultatif : La balise VLAN. Aucune configuration VLAN supplémentaire n'est requise au niveau de la stratégie de configuration du réseau de nœuds.

NOTE

La définition de l'attachement au réseau d'un pont Linux est la méthode la plus efficace pour connecter une machine virtuelle à un VLAN.

3. Créer la définition de la pièce jointe au réseau :

```
oc create -f <network-attachment-definition.yaml> 1
```

1. Où **<network-attachment-definition.yaml>** est le nom de fichier du manifeste de définition des pièces jointes au réseau.

Vérification

- Vérifiez que la définition de la pièce jointe au réseau a été créée en exécutant la commande suivante :

```
oc get network-attachment-definition <bridge-network>
```

10.21.3.3. Configuration de la machine virtuelle pour un réseau pont Linux

10.21.3.3.1. Créer un NIC pour une machine virtuelle dans la console web

Créez et attachez des cartes réseau supplémentaires à une machine virtuelle à partir de la console web.

Conditions préalables

- Une définition de l'attachement au réseau doit être disponible.

Procédure

1. Dans le bon projet de la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Network Interfaces** pour afficher les cartes réseau déjà connectées à la machine virtuelle.
4. Cliquez sur **Add Network Interface** pour créer un nouvel emplacement dans la liste.
5. Sélectionnez une définition d'attachement au réseau dans la liste **Network** pour le réseau supplémentaire.
6. Remplissez les champs **Name**, **Model**, **Type**, et **MAC Address** pour le nouveau NIC.
7. Cliquez sur **Save** pour enregistrer et attacher le NIC à la machine virtuelle.

10.21.3.3.2. Domaines de mise en réseau

Nom	Description
Nom	Nom du contrôleur d'interface réseau.
Model	Indique le modèle du contrôleur d'interface réseau. Les valeurs prises en charge sont **e1000e** et **virtio**.
Réseau	Liste des définitions de pièces jointes disponibles.
Type	Liste des méthodes de liaison disponibles. Sélectionnez la méthode de liaison adaptée à l'interface réseau : • Réseau de pods par défaut : **masquerade** • Réseau de ponts Linux : **bridge** • Réseau SR-IOV : **SR-IOV**

Nom	Description
Adresse MAC	Adresse MAC du contrôleur d'interface réseau. Si aucune adresse MAC n'est spécifiée, une adresse est attribuée automatiquement.

10.21.3.3.3. Attacher une machine virtuelle à un réseau supplémentaire dans le CLI

Attachez une machine virtuelle à un réseau supplémentaire en ajoutant une interface de pont et en spécifiant une définition d'attachement réseau dans la configuration de la machine virtuelle.

Cette procédure utilise un fichier YAML pour démontrer l'édition de la configuration et l'application du fichier mis à jour au cluster. Vous pouvez également utiliser la commande **oc edit <object> <name>** pour modifier une machine virtuelle existante.

Conditions préalables

- Arrêtez la machine virtuelle avant de modifier la configuration. Si vous modifiez une machine virtuelle en cours d'exécution, vous devez la redémarrer pour que les modifications soient prises en compte.

Procédure

1. Créez ou modifiez la configuration d'une machine virtuelle que vous souhaitez connecter au réseau de ponts.

2. Ajoutez l'interface de pont à la liste **spec.template.spec.domain.devices.interfaces** et la définition de l'attachement au réseau à la liste **spec.template.spec.networks**. Cet exemple ajoute une interface de pont appelée **bridge-net** qui se connecte à la définition de l'attachement au réseau **a-bridge-network**:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
    name: <example-vm>
spec:
  template:
    spec:
      domain:
        devices:
          interfaces:
            - masquerade: {}
              name: <default>
            - bridge: {}
              name: <bridge-net> 1
...
      networks:
        - name: <default>
          pod: {}
        - name: <bridge-net> 2
```

```
        multus:
          networkName: <network-namespace>/<a-bridge-network> 3
...
```

1. Le nom de l'interface du pont.
2. Le nom du réseau. Cette valeur doit correspondre à la valeur **name** de l'entrée **spec.template.spec.domain.devices.interfaces** correspondante.
3. Le nom de la définition de l'attachement réseau, préfixé par l'espace de noms dans lequel il existe. L'espace de noms doit être soit l'espace de noms **default**, soit l'espace de noms dans lequel la VM doit être créée. Dans ce cas, **multus** est utilisé. Multus est un plugin d'interface de réseau en nuage (CNI) qui permet à plusieurs CNI d'exister afin qu'un pod ou une machine virtuelle puisse utiliser les interfaces dont il a besoin.

3. Appliquer la configuration :

```
oc apply -f <example-vm.yaml>
```

4. Facultatif : si vous modifiez une machine virtuelle en cours d'exécution, vous devez la redémarrer pour que les modifications soient prises en compte.

10.21.4. Connexion d'une machine virtuelle à un réseau SR-IOV

Vous pouvez connecter une machine virtuelle (VM) à un réseau de virtualisation d'E/S à racine unique (SR-IOV) en procédant comme suit :

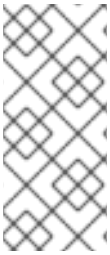
1. Configurer un périphérique de réseau SR-IOV.
2. Configurer un réseau SR-IOV.
3. Connecter la VM au réseau SR-IOV.

10.21.4.1. Conditions préalables

- Vous devez avoir activé les paramètres SR-IOV et VT-d globaux dans le micrologiciel de l'hôte .
- Vous devez avoir installé l'opérateur de réseau SR-IOV.

10.21.4.2. Configuration des périphériques de réseau SR-IOV

L'opérateur de réseau SR-IOV ajoute la ressource **SriovNetworkNodePolicy.sriovnetwork.openshift.io** CustomResourceDefinition à OpenShift Container Platform. Vous pouvez configurer un périphérique réseau SR-IOV en créant une ressource personnalisée (CR) SriovNetworkNodePolicy.



NOTE

Lors de l'application de la configuration spécifiée dans un objet **SriovNetworkNodePolicy**, l'opérateur SR-IOV peut vidanger les nœuds et, dans certains cas, les redémarrer.

L'application d'une modification de configuration peut prendre plusieurs minutes.

Conditions préalables

- You installed the OpenShift CLI (**oc**).
- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez installé l'opérateur de réseau SR-IOV.
- Vous avez suffisamment de nœuds disponibles dans votre cluster pour gérer la charge de travail expulsée des nœuds épuisés.
- Vous n'avez sélectionné aucun nœud du plan de contrôle pour la configuration des équipements du réseau SR-IOV.

Procédure

1. Créez un objet **SriovNetworkNodePolicy**, puis enregistrez le YAML dans le fichier **<name>-sriov-node-network.yaml**. Remplacez **<name>** par le nom de cette configuration.

```
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: <name> 1
  namespace: openshift-sriov-network-operator 2
spec:
  resourceName: <sriov_resource_name> 3
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true" 4
  priority: <priority> 5
  mtu: <mtu> 6
  numVfs: <num> 7
  nicSelector: 8
    vendor: "<vendor_code>" 9
    deviceID: "<device_id>" 10
    pfNames: ["<pf_name>", ...] 11
    rootDevices: ["<pci_bus_id>", "..."] 12
  deviceType: vfio-pci 13
  isRdma: false 14
```

1. Spécifiez un nom pour l'objet CR.
2. Indiquer l'espace de noms dans lequel l'opérateur SR-IOV est installé.
3. Indiquez le nom de la ressource du plugin de périphérique SR-IOV. Vous pouvez créer plusieurs objets **SriovNetworkNodePolicy** pour un nom de ressource.
4. Spécifiez le sélecteur de nœuds pour sélectionner les nœuds à configurer. Seuls les périphériques réseau SR-IOV des nœuds sélectionnés sont configurés. Le plugin SR-IOV Container Network Interface (CNI) et le plugin de périphérique sont déployés uniquement sur les nœuds sélectionnés.
5. Facultatif : Indiquez une valeur entière comprise entre **0** et **99**. Un nombre plus petit a une priorité plus élevée, donc une priorité de **10** est plus élevée qu'une priorité de **99**. La valeur par défaut est **99**.

6 Facultatif : Spécifiez une valeur pour l'unité de transmission maximale (MTU) de la fonction virtuelle. La valeur maximale du MTU peut varier selon les modèles de NIC.

7 Indiquez le nombre de fonctions virtuelles (VF) à créer pour le périphérique de réseau physique SR-IOV. Pour un contrôleur d'interface réseau (NIC) Intel, le nombre de VF ne peut pas être supérieur au nombre total de VF pris en charge par le périphérique. Pour un NIC Mellanox, le nombre de VF ne peut pas être supérieur à **128**.

8 Le mappage **nicSelector** sélectionne l'appareil Ethernet que l'opérateur doit configurer. Il n'est pas nécessaire de spécifier des valeurs pour tous les paramètres. Il est recommandé d'identifier l'adaptateur Ethernet avec suffisamment de précision pour minimiser la possibilité de sélectionner un périphérique Ethernet par inadvertance. Si vous spécifiez **rootDevices**, vous devez également spécifier une valeur pour **vendor**, **deviceID** ou **pfNames**. Si vous spécifiez **pfNames** et **rootDevices** en même temps, assurez-vous qu'ils pointent vers un périphérique identique.

9 Facultatif : Indiquez le code hexadécimal du fournisseur de l'appareil réseau SR-IOV. Les seules valeurs autorisées sont **8086** ou **15b3**.

10 Facultatif : Indiquez le code hexadécimal du périphérique du réseau SR-IOV. Les seules valeurs autorisées sont **158b**, **1015**, **1017**.

11 Facultatif : Ce paramètre accepte un tableau d'un ou plusieurs noms de fonctions physiques (PF) pour le périphérique Ethernet.

12 Ce paramètre accepte un tableau d'une ou plusieurs adresses de bus PCI pour la fonction physique du périphérique Ethernet. Indiquez l'adresse au format suivant : **0000:02:00.1**.

13 Le type de pilote **vfio-pci** est requis pour les fonctions virtuelles dans OpenShift Virtualization.

14 Facultatif : Indiquez si le mode RDMA (remote direct memory access) doit être activé. Pour une carte Mellanox, définissez **isRdma** sur **false**. La valeur par défaut est **false**.

NOTE

Si l'indicateur **isRDMA** est défini sur **true**, vous pouvez continuer à utiliser le VF compatible RDMA comme un périphérique réseau normal. Un périphérique peut être utilisé dans l'un ou l'autre mode.

2. Facultatif : Étiqueter les nœuds de cluster compatibles SR-IOV avec **SriovNetworkNodePolicy.Spec.NodeSelector** s'ils ne le sont pas déjà. Pour plus d'informations sur l'étiquetage des nœuds, voir "Understanding how to update labels on nodes".

3. Créer l'objet **SriovNetworkNodePolicy**:

```
oc create -f <name>-sriov-node-network.yaml
```

où **<name>** spécifie le nom de cette configuration.

Après l'application de la mise à jour de la configuration, tous les pods de l'espace de noms **sriov-network-operator** passent à l'état **Running**.

4. Pour vérifier que le dispositif de réseau SR-IOV est configuré, entrez la commande suivante. Remplacez **<node_name>** par le nom d'un nœud avec le dispositif de réseau SR-IOV que vous venez de configurer.

```
oc get sriovnetworknodestates -n openshift-sriov-network-operator <node_name> -o
jsonpath='{.status.syncStatus}'
```

10.21.4.3. Configuration du réseau supplémentaire SR-IOV

Vous pouvez configurer un réseau supplémentaire qui utilise le matériel SR-IOV en créant un objet **SriovNetwork**.

Lorsque vous créez un objet **SriovNetwork**, l'opérateur de réseau SR-IOV crée automatiquement un objet **NetworkAttachmentDefinition**.

NOTE

Ne modifiez pas et ne supprimez pas un objet **SriovNetwork** s'il est attaché à des pods ou à des machines virtuelles dans un état **running**.

Conditions préalables

- Installez le CLI OpenShift (**oc**).
- Connectez-vous en tant qu'utilisateur disposant des privilèges **cluster-admin**.

Procédure

1. Créez l'objet **SriovNetwork** suivant, puis enregistrez le YAML dans le fichier **<name>-sriov-network.yaml**. Remplacez **<name>** par un nom pour ce réseau supplémentaire.

```
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetwork
metadata:
  name: <name> 1
  namespace: openshift-sriov-network-operator 2
spec:
  resourceName: <sriov_resource_name> 3
  networkNamespace: <target_namespace> 4
  vlan: <vlan> 5
  spoofChk: "<spoof_check>" 6
  linkState: <link_state> 7
  maxTxRate: <max_tx_rate> 8
  minTxRate: <min_rx_rate> 9
  vlanQoS: <vlan_qos> 10
  trust: "<trust_vf>" 11
  capabilities: <capabilities> 12
```

1. Remplacez **<name>** par un nom pour l'objet. L'opérateur du réseau SR-IOV crée un objet **NetworkAttachmentDefinition** portant le même nom.
2. Indiquer l'espace de noms dans lequel l'opérateur de réseau SR-IOV est installé.

3. Remplacer **<sriov_resource_name>** par la valeur du paramètre **.spec.resourceName** de l'objet **SriovNetworkNodePolicy** qui définit le matériel SR-IOV pour ce réseau supplémentaire.

4. Remplacez **<target_namespace>** par l'espace de noms cible du SriovNetwork. Seuls les pods ou les machines virtuelles de l'espace de noms cible peuvent s'attacher au SriovNetwork.

5. Facultatif : Remplacez **<vlan>** par un ID de réseau local virtuel (VLAN) pour le réseau supplémentaire. La valeur entière doit être comprise entre **0** et **4095**. La valeur par défaut est **0**.

6. Facultatif : Remplacer **<spoof_check>** par le mode de vérification de l'usurpation d'identité du VF. Les valeurs autorisées sont les chaînes **"on"** et **"off"**.

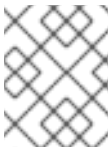
IMPORTANT

Vous devez mettre la valeur que vous indiquez entre guillemets, sinon le CR est rejeté par l'opérateur de réseau SR-IOV.

7. Facultatif : Remplacer **<link_state>** par l'état de la liaison de la fonction virtuelle (VF). Les valeurs autorisées sont **enable**, **disable** et **auto**.

8. Facultatif : Remplacez **<max_tx_rate>** par un taux de transmission maximal, en Mbps, pour le VF.

9. Facultatif : Remplacer **<min_tx_rate>** par un taux de transmission minimum, en Mbps, pour le VF. Cette valeur doit toujours être inférieure ou égale à la vitesse de transmission maximale.



NOTE

Les cartes réseau Intel ne prennent pas en charge le paramètre **minTxRate**. Pour plus d'informations, voir BZ#1772847.

10. Facultatif : Remplacez **<vlan_qos>** par un niveau de priorité IEEE 802.1p pour le VF. La valeur par défaut est **0**.

11. Optionnel : Remplacer **<trust_vf>** par le mode de confiance du VF. Les valeurs autorisées sont les chaînes **"on"** et **"off"**.

IMPORTANT

Vous devez mettre la valeur que vous indiquez entre guillemets, sinon le CR est rejeté par l'opérateur de réseau SR-IOV.

12. Facultatif : Remplacez **<capabilities>** par les capacités à configurer pour ce réseau.

2. Pour créer l'objet, entrez la commande suivante. Remplacez **<name>** par un nom pour ce réseau supplémentaire.

```
oc create -f <name>-sriov-network.yaml
```

3. Facultatif : Pour confirmer que l'objet **NetworkAttachmentDefinition** associé à l'objet **SriovNetwork** que vous avez créé à l'étape précédente existe, entrez la commande suivante. Remplacez **<namespace>** par l'espace de noms que vous avez spécifié dans l'objet **SriovNetwork**.

```
oc get net-attach-def -n <namespace>
```

10.21.4.4. Connexion d'une machine virtuelle à un réseau SR-IOV

Vous pouvez connecter la machine virtuelle (VM) au réseau SR-IOV en incluant les détails du réseau dans la configuration de la VM.

Procédure

1. Inclure les détails du réseau SR-IOV dans les pages **spec.domain.devices.interfaces** et **spec.networks** de la configuration de la VM :

```
kind: VirtualMachine
...
spec:
  domain:
    devices:
      interfaces:
      - name: <default> 1
        masquerade: {} 2
      - name: <nic1> 3
        sriov: {}
  networks:
  - name: <default> 4
    pod: {}
  - name: <nic1> 5
    multus:
        networkName: <sriov-network> 6
...
```

1. Un nom unique pour l'interface qui est connectée au réseau de pods.
2. La liaison **masquerade** au réseau de pods par défaut.
3. Un nom unique pour l'interface SR-IOV.
4. Le nom de l'interface réseau du pod. Ce nom doit être identique à celui de **interfaces.name** que vous avez défini précédemment.
5. Le nom de l'interface SR-IOV. Il doit être identique au nom **interfaces.name** que vous avez défini précédemment.
6. Le nom de la définition de l'attachement au réseau SR-IOV.

2. Appliquer la configuration de la machine virtuelle :

```
oc apply -f <vm-sriov.yaml> 1
```

1. Le nom du fichier YAML de la machine virtuelle.

10.21.5. Connexion d'une machine virtuelle à un maillage de services

OpenShift Virtualization est désormais intégré à OpenShift Service Mesh. Vous pouvez surveiller, visualiser et contrôler le trafic entre les pods qui exécutent des charges de travail de machines virtuelles sur le réseau de pods par défaut avec IPv4.

10.21.5.1. Conditions préalables

- Vous devez avoir installé le Service Mesh Operator et déployé le plan de contrôle du service mesh.
- Vous devez avoir ajouté l'espace de noms où la machine virtuelle est créée au rouleau de membres du maillage de services.
- Vous devez utiliser la méthode de liaison **masquerade** pour le réseau de pods par défaut.

10.21.5.2. Configuration d'une machine virtuelle pour le maillage de services

Pour ajouter une charge de travail de machine virtuelle (VM) à un maillage de services, activez l'injection automatique de sidecar dans le fichier de configuration de la VM en définissant l'annotation **sidecar.istio.io/inject** sur **true**. Exposez ensuite votre VM en tant que service pour visualiser votre application dans le maillage.

Conditions préalables

- Pour éviter les conflits de ports, n'utilisez pas les ports utilisés par le proxy Istio sidecar. Il s'agit notamment des ports 15000, 15001, 15006, 15008, 15020, 15021 et 15090.

Procédure

1. Modifiez le fichier de configuration de la VM pour ajouter l'annotation **sidecar.istio.io/inject: "true"**.

Exemple de fichier de configuration

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    kubevirt.io/vm: vm-istio
  name: vm-istio
spec:
  runStrategy: Always
  template:
    metadata:
      labels:
        kubevirt.io/vm: vm-istio
        app: vm-istio (1)
      annotations:
        sidecar.istio.io/inject: "true" (2)
    spec:
      domain:
        devices:
          interfaces:
          - name: default
            masquerade: {} (3)
          disks:
```

```
      - disk:
          bus: virtio
        name: containerdisk
      - disk:
          bus: virtio
        name: cloudinitdisk
    resources:
      requests:
        memory: 1024M
  networks:
  - name: default
    pod: {}
  terminationGracePeriodSeconds: 180
  volumes:
  - containerDisk:
      image: registry:5000/kubevirt/fedora-cloud-container-disk-demo:devel
    name: containerdisk
```

1. La paire clé/valeur (étiquette) qui doit être associée à l'attribut du sélecteur de service.
2. L'annotation permettant d'activer l'injection automatique de sidecar.
3. La méthode de liaison (mode masqué) à utiliser avec le réseau de pods par défaut.

2. Appliquer la configuration de la VM :

```
oc apply -f <vm_name>.yaml 1
```

1. Le nom du fichier YAML de la machine virtuelle.

3. Créez un objet **Service** pour exposer votre VM au réseau de services.

```
apiVersion: v1
kind: Service
metadata:
  name: vm-istio
spec:
  selector:
    app: vm-istio 1
  ports:
    - port: 8080
      name: http
      protocol: TCP
```

1. Le sélecteur de service qui détermine l'ensemble des pods ciblés par un service. Cet attribut correspond au champ **spec.metadata.labels** dans le fichier de configuration de la VM. Dans l'exemple ci-dessus, l'objet **Service** nommé **vm-istio** cible le port TCP 8080 sur n'importe quel module portant l'étiquette **app=vm-istio**.

4. Créer le service :

```
oc create -f <service_name>.yaml 1
```

1 Le nom du fichier YAML du service.

10.21.6. Configuration des adresses IP pour les machines virtuelles

Vous pouvez configurer des adresses IP dynamiques ou statiques pour les machines virtuelles.

Conditions préalables

- La machine virtuelle doit se connecter à un réseau externe.
- Vous devez disposer d'un serveur DHCP sur le réseau supplémentaire pour configurer une IP dynamique pour la machine virtuelle.

10.21.6.1. Configurer une adresse IP pour une nouvelle machine virtuelle à l'aide de cloud-init

Vous pouvez utiliser cloud-init pour configurer une adresse IP lorsque vous créez une machine virtuelle. L'adresse IP peut être fournie de manière dynamique ou statique.

Procédure

- Créez une configuration de machine virtuelle et incluez les détails du réseau cloud-init dans le champ **spec.volumes.cloudInitNoCloud.networkData** de la configuration de la machine virtuelle :

  a. Pour configurer une IP dynamique, spécifiez le nom de l'interface et le booléen **dhcp4**:

```
kind: VirtualMachine
spec:
...
  volumes:
  - cloudInitNoCloud:
      networkData: |
        version: 2
        ethernets:
          eth1: 1
            dhcp4: true 2
```

1 Le nom de l'interface.

2 Utilise DHCP pour fournir une adresse IPv4.

  b. Pour configurer une IP statique, spécifiez le nom de l'interface et l'adresse IP :

```
kind: VirtualMachine
spec:
...
  volumes:
  - cloudInitNoCloud:
      networkData: |
        version: 2
        ethernets:
```

```
eth1: 1
  addresses:
  - 10.10.10.14/24 2
```

1 Le nom de l'interface.

2 L'adresse IP statique de la machine virtuelle.

10.21.7. Afficher l'adresse IP des cartes réseau d'une machine virtuelle

Vous pouvez afficher l'adresse IP d'un contrôleur d'interface réseau (NIC) en utilisant la console Web ou le client **oc**. L'agent invité QEMU affiche des informations supplémentaires sur les réseaux secondaires de la machine virtuelle.

10.21.7.1. Conditions préalables

- Installer l'agent invité QEMU sur la machine virtuelle.

10.21.7.2. Afficher l'adresse IP d'une interface de machine virtuelle dans le CLI

La configuration de l'interface réseau est incluse dans la commande **oc describe vmi <vmi_name>**.

Vous pouvez également afficher les informations relatives à l'adresse IP en exécutant **ip addr** sur la machine virtuelle ou en exécutant **oc get vmi <vmi_name> -o yaml**.

Procédure

- Utilisez la commande **oc describe** pour afficher la configuration de l'interface de la machine virtuelle :

```
$ oc describe vmi <nom_du_vmi>
```

Exemple de sortie

```
...
Interfaces:
   Interface Name:  eth0
   Ip Address:      10.244.0.37/24
   Ip Addresses:
     10.244.0.37/24
     fe80::858:aff:fef4:25/64
   Mac:             0a:58:0a:f4:00:25
   Name:            default
   Interface Name:  v2
   Ip Address:      1.1.1.7/24
   Ip Addresses:
     1.1.1.7/24
     fe80::f4d9:70ff:fe13:9089/64
   Mac:             f6:d9:70:13:90:89
   Interface Name:  v1
   Ip Address:      1.1.1.1/24
   Ip Addresses:
     1.1.1.1/24
```

```
1.1.1.2/24
1.1.1.4/24
2001:de7:0:f101::1/64
2001:db8:0:f101::1/64
fe80::1420:84ff:fe10:17aa/64
Mac:            16:20:84:10:17:aa
```

10.21.7.3. Afficher l'adresse IP d'une interface de machine virtuelle dans la console web

Les informations IP sont affichées sur la page **VirtualMachine details** pour la machine virtuelle.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez un nom de machine virtuelle pour ouvrir la page **VirtualMachine details**.

Les informations relatives à chaque NIC connecté sont affichées sous **IP Address** dans l'onglet **Details**.

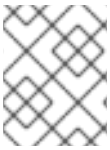
10.21.8. Utilisation d'un pool d'adresses MAC pour les machines virtuelles

Le composant *KubeMacPool* fournit un service de pool d'adresses MAC pour les cartes d'interface réseau des machines virtuelles dans un espace de noms.

10.21.8.1. À propos de KubeMacPool

KubeMacPool fournit un pool d'adresses MAC par espace de noms et attribue des adresses MAC aux cartes d'interface réseau des machines virtuelles à partir du pool. Cela garantit que la carte d'interface réseau se voit attribuer une adresse MAC unique qui n'entre pas en conflit avec l'adresse MAC d'une autre machine virtuelle.

Les instances de machines virtuelles créées à partir de cette machine virtuelle conservent l'adresse MAC attribuée lors des redémarrages.



NOTE

KubeMacPool ne gère pas les instances de machines virtuelles créées indépendamment d'une machine virtuelle.

KubeMacPool est activé par défaut lors de l'installation d'OpenShift Virtualization. Vous pouvez désactiver un pool d'adresses MAC pour un espace de noms en ajoutant le label **mutatevirtualmachines.kubemacpool.io=ignore** à l'espace de noms. Réactivez KubeMacPool pour l'espace de noms en supprimant le label.

10.21.8.2. Désactivation d'un pool d'adresses MAC pour un espace de noms dans la CLI

Désactiver un pool d'adresses MAC pour les machines virtuelles dans un espace de noms en ajoutant l'étiquette **mutatevirtualmachines.kubemacpool.io=ignore** à l'espace de noms.

Procédure

- Ajoutez l'étiquette **mutatevirtualmachines.kubemacpool.io=ignore** à l'espace de noms. L'exemple suivant désactive KubeMacPool pour deux espaces de noms, **<namespace1>** et **<namespace2>**:

```
$ oc label namespace <namespace1> <namespace2>
mutatevirtualmachines.kubemacpool.io=ignore
```

10.21.8.3. Réactivation d'un pool d'adresses MAC pour un espace de noms dans la CLI

Si vous avez désactivé KubeMacPool pour un espace de noms et que vous souhaitez le réactiver, supprimez l'étiquette **mutatevirtualmachines.kubemacpool.io=ignore** de l'espace de noms.

NOTE

Les versions précédentes d'OpenShift Virtualization utilisaient le label **mutatevirtualmachines.kubemacpool.io=allocate** pour activer KubeMacPool pour un espace de noms. Ceci est toujours supporté mais redondant car KubeMacPool est maintenant activé par défaut.

Procédure

- Supprimer l'étiquette KubeMacPool de l'espace de noms. L'exemple suivant réactive KubeMacPool pour deux espaces de noms, **<namespace1>** et **<namespace2>**:

```
oc label namespace <namespace1> <namespace2> mutatevirtualmachines.kubemacpool.io-
```

10.22. DISQUES DE LA MACHINE VIRTUELLE

10.22.1. Caractéristiques de stockage

Utilisez le tableau suivant pour déterminer la disponibilité des fonctionnalités pour le stockage persistant local et partagé dans OpenShift Virtualization.

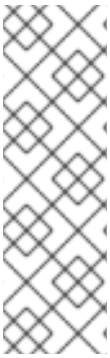
10.22.1.1. Matrice des fonctionnalités de stockage d'OpenShift Virtualization

Tableau 10.5. Matrice des fonctionnalités de stockage d'OpenShift Virtualization

	Migration en direct de la machine virtuelle	Clonage de disques de machines virtuelles assisté par l'hôte	Clonage de disques de machines virtuelles assisté par le stockage	Instantanés de machines virtuelles
OpenShift Data Foundation : Volumes en mode bloc RBD	Oui	Oui	Oui	Oui
OpenShift Virtualization hostpath provisioner	Non	Oui	Non	Non

	Migration en direct de la machine virtuelle	Clonage de disques de machines virtuelles assisté par l'hôte	Clonage de disques de machines virtuelles assisté par le stockage	Instantanés de machines virtuelles
Autre système de stockage inscriptible à plusieurs nœuds	Oui [1]	Oui	Oui [2]	Oui [2]
Autre système de stockage inscriptible à nœud unique	Non	Oui	Oui [2]	Oui [2]

1. Les PVC doivent demander un mode d'accès ReadWriteMany.
2. Le fournisseur de stockage doit prendre en charge les API d'instantanés Kubernetes et CSI



NOTE

Vous ne pouvez pas migrer en direct les machines virtuelles qui utilisent :

- Une classe de stockage avec un mode d'accès ReadWriteOnce (RWO)
- Fonctionnalités passthrough telles que les GPU

Ne définissez pas le champ **evictionStrategy** sur **LiveMigrate** pour ces machines virtuelles.

10.22.2. Configuration du stockage local pour les machines virtuelles

Vous pouvez configurer le stockage local pour les machines virtuelles en utilisant le Hostpath Provisioner (HPP).

Lorsque vous installez l'opérateur de virtualisation OpenShift, l'opérateur Hostpath Provisioner (HPP) est automatiquement installé. Le HPP est un provisionneur de stockage local conçu pour OpenShift Virtualization et créé par l'opérateur Hostpath Provisioner. Pour utiliser le HPP, vous devez créer une ressource personnalisée (CR) HPP.

10.22.2.1. Création d'un provisionneur de chemins d'accès avec un pool de stockage de base

Vous configurez un hostpath provisioner (HPP) avec un pool de stockage de base en créant une ressource personnalisée HPP (CR) avec une strophe **storagePools**. Le pool de stockage spécifie le nom et le chemin d'accès utilisés par le pilote CSI.

Conditions préalables

- Les répertoires spécifiés dans **spec.storagePools.path** doivent être accessibles en lecture/écriture.
- Les pools de stockage ne doivent pas se trouver dans la même partition que le système d'exploitation. Sinon, la partition du système d'exploitation risque d'être saturée, ce qui aura un impact sur les performances ou rendra le nœud instable ou inutilisable.

Procédure

1. Créez un fichier **hpp_cr.yaml** avec une strophe **storagePools** comme dans l'exemple suivant :

```
apiVersion: hostpathprovisioner.kubevirt.io/v1beta1
kind: HostPathProvisioner
metadata:
  name: hostpath-provisioner
spec:
  imagePullPolicy: IfNotPresent
  storagePools: 1
  - name: any_name
    path: "/var/myvolumes" 2
workload:
  nodeSelector:
    kubernetes.io/os: linux
```

1. La strophe **storagePools** est un tableau auquel vous pouvez ajouter plusieurs entrées.
2. Spécifiez les répertoires du pool de stockage sous le chemin de ce nœud.

2. Enregistrez le fichier et quittez.
3. Créez le HPP en exécutant la commande suivante :

```
$ oc create -f hpp_cr.yaml
```

10.22.2.1.1. Création de classes de stockage

Lorsque vous créez une classe de stockage, vous définissez des paramètres qui affectent le provisionnement dynamique des volumes persistants (PV) appartenant à cette classe de stockage. Vous ne pouvez pas mettre à jour les paramètres d'un objet **StorageClass** après l'avoir créé.

Pour utiliser le hostpath provisioner (HPP), vous devez créer une classe de stockage associée pour le pilote CSI avec la strophe **storagePools**.

NOTE

Les machines virtuelles utilisent des volumes de données basés sur des PV locaux. Les PV locaux sont liés à des nœuds spécifiques. Lorsque l'image de disque est préparée pour être consommée par la machine virtuelle, il est possible que la machine virtuelle ne puisse pas être planifiée sur le nœud où la PV de stockage local a été précédemment épinglée.

Pour résoudre ce problème, utilisez le planificateur de pods Kubernetes pour lier la réclamation de volume persistant (PVC) à un PV sur le bon nœud. En utilisant la valeur **StorageClass** avec le paramètre **volumeBindingMode** défini sur **WaitForFirstConsumer**, la liaison et le provisionnement de la PV sont retardés jusqu'à ce qu'un pod soit créé à l'aide du PVC.

10.22.2.1.2. Création d'une classe de stockage pour le pilote CSI avec la strophe storagePools

Vous créez une ressource personnalisée (CR) de classe de stockage pour le pilote CSI de Hostpath Provisioner (HPP).

Procédure

1. Créez un fichier **storageclass_csi.yaml** pour définir la classe de stockage :

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath-csi
provisioner: kubevirt.io.hostpath-provisioner
reclaimPolicy: Delete (1)
volumeBindingMode: WaitForFirstConsumer (2)
parameters:
  storagePool: my-storage-pool (3)
```

(1) Les deux valeurs possibles de **reclaimPolicy** sont **Delete** et **Retain**. Si vous ne spécifiez pas de valeur, la valeur par défaut est **Delete**.

(2) Le paramètre **volumeBindingMode** détermine le moment où le provisionnement dynamique et la liaison de volume se produisent. Spécifiez **WaitForFirstConsumer** pour retarder la liaison et le provisionnement d'un volume persistant (PV) jusqu'à ce qu'un pod utilisant la réclamation de volume persistant (PVC) soit créé. Cela permet de s'assurer que le PV répond aux exigences d'ordonnancement du pod.

(3) Indiquez le nom du pool de stockage défini dans le CR HPP.

1. Enregistrez le fichier et quittez.
2. Créez l'objet **StorageClass** en exécutant la commande suivante :

```
$ oc create -f storageclass_csi.yaml
```

10.22.2.2. À propos des pools de stockage créés avec des modèles PVC

Si vous disposez d'un seul volume persistant (PV) de grande taille, vous pouvez créer un pool de stockage en définissant un modèle PVC dans la ressource personnalisée (CR) du Hostpath Provisioner (HPP).

Un pool de stockage créé avec un modèle PVC peut contenir plusieurs volumes HPP. La division d'un PV en volumes plus petits offre une plus grande flexibilité pour l'allocation des données.

Le modèle PVC est basé sur la strophe **spec** de l'objet **PersistentVolumeClaim**:

Exemple d'objet PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: iso-pvc
spec:
  volumeMode: Block (1)
  storageClassName: my-storage-class
  accessModes:
  - ReadWriteOnce
```

```
resources:
  requests:
    storage: 5Gi
```

1 Cette valeur n'est requise que pour les PV en mode volume par bloc.

Vous définissez un pool de stockage à l'aide d'une spécification **pvcTemplate** dans le CR HPP. L'opérateur crée un PVC à partir de la spécification **pvcTemplate** pour chaque nœud contenant le pilote HPP CSI. Le PVC créé à partir du modèle de PVC consomme le grand PV unique, ce qui permet au HPP de créer des volumes dynamiques plus petits.

Vous pouvez combiner des pools de stockage de base avec des pools de stockage créés à partir de modèles PVC.

10.22.2.2.1. Création d'un pool de stockage avec un modèle PVC

Vous pouvez créer un pool de stockage pour plusieurs volumes HPP (Hostpath Provisioner) en spécifiant un modèle PVC dans la ressource personnalisée HPP (CR).

Conditions préalables

- Les répertoires spécifiés dans **spec.storagePools.path** doivent être accessibles en lecture/écriture.
- Les pools de stockage ne doivent pas se trouver dans la même partition que le système d'exploitation. Sinon, la partition du système d'exploitation risque d'être saturée, ce qui aura un impact sur les performances ou rendra le nœud instable ou inutilisable.

Procédure

1. Créez un fichier **hpp_pvc_template_pool.yaml** pour le CR HPP qui spécifie un modèle de volume persistant (PVC) dans la strophe **storagePools** selon l'exemple suivant :

```
apiVersion: hostpathprovisioner.kubevirt.io/v1beta1
kind: HostPathProvisioner
metadata:
  name: hostpath-provisioner
spec:
  imagePullPolicy: IfNotPresent
  storagePools: 1
  - name: my-storage-pool
    path: "/var/myvolumes" 2
    pvcTemplate:
      volumeMode: Block 3
      storageClassName: my-storage-class 4
      accessModes:
      - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi 5
  workload:
    nodeSelector:
      kubernetes.io/os: linux
```

1. Le stanza **storagePools** est un tableau qui peut contenir des pools de stockage de base et des pools de stockage de modèles PVC.
2. Spécifiez les répertoires du pool de stockage sous le chemin de ce nœud.
3. Facultatif : Le paramètre **volumeMode** peut être **Block** ou **Filesystem** tant qu'il correspond au format du volume provisionné. Si aucune valeur n'est spécifiée, la valeur par défaut est **Filesystem**. Si **volumeMode** est **Block**, le pod de montage crée un système de fichiers XFS sur le volume bloc avant de le monter.
4. Si le paramètre **storageClassName** est omis, la classe de stockage par défaut est utilisée pour créer des PVC. Si vous omettez **storageClassName**, assurez-vous que la classe de stockage HPP n'est pas la classe de stockage par défaut.
5. Vous pouvez spécifier un stockage provisionné statiquement ou dynamiquement. Dans les deux cas, assurez-vous que la taille de stockage demandée est appropriée pour le volume que vous souhaitez diviser virtuellement ou que le PVC ne peut pas être lié au grand PV. Si la classe de stockage que vous utilisez utilise un stockage à provisionnement dynamique, choisissez une taille d'allocation qui correspond à la taille d'une demande typique.

2. Enregistrez le fichier et quittez.
3. Créez le HPP avec un pool de stockage en exécutant la commande suivante :

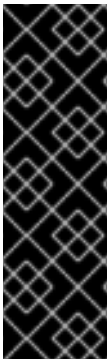
```
$ oc create -f hpp_pvc_template_pool.yaml
```

Ressources supplémentaires

- Personnalisation du profil de stockage

10.22.3. Création de volumes de données

Vous pouvez créer un volume de données à l'aide de l'API PVC ou de l'API de stockage.



IMPORTANT

Lors de l'utilisation d'OpenShift Virtualization avec OpenShift Container Platform Container Storage, spécifiez des réclamations de volumes persistants (PVC) en mode bloc RBD lors de la création de disques de machines virtuelles. Avec les disques de machines virtuelles, les volumes en mode bloc RBD sont plus efficaces et offrent de meilleures performances que les PVC Ceph FS ou RBD en mode système de fichiers.

Pour spécifier les PVC en mode bloc RBD, utilisez la classe de stockage 'ocs-storagecluster-ceph-rbd' et **VolumeMode: Block**.

ASTUCE

Dans la mesure du possible, utilisez l'API de stockage pour optimiser l'allocation d'espace et maximiser les performances.

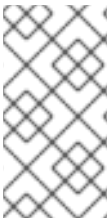
Un site *storage profile* est une ressource personnalisée gérée par le CDI. Il fournit des paramètres de stockage recommandés en fonction de la classe de stockage associée. Un profil de stockage est attribué à chaque classe de stockage.

Les profils de stockage vous permettent de créer rapidement des volumes de données tout en réduisant le codage et en minimisant les erreurs potentielles.

Pour les types de stockage reconnus, le CDI fournit des valeurs qui optimisent la création de PVC. Cependant, vous pouvez configurer des paramètres automatiques pour une classe de stockage si vous personnalisez le profil de stockage.

10.22.3.1. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.22.3.2. Création de volumes de données à l'aide de l'API de stockage

Lorsque vous créez un volume de données à l'aide de l'API de stockage, l'interface de données conteneurisées (CDI) optimise l'allocation de votre revendication de volume persistant (PVC) en fonction du type de stockage pris en charge par la classe de stockage sélectionnée. Il vous suffit de spécifier le nom du volume de données, l'espace de noms et la quantité de stockage que vous souhaitez allouer.

Par exemple :

- Lors de l'utilisation de Ceph RBD, **accessModes** est automatiquement défini sur **ReadWriteMany**, ce qui permet une migration en direct. **volumeMode** est défini sur **Block** pour maximiser les performances.

- Lorsque vous utilisez **volumeMode: Filesystem**, le CDI demandera automatiquement plus d'espace, si nécessaire, pour tenir compte de l'encombrement du système de fichiers.

Dans le fichier YAML suivant, l'utilisation de l'API de stockage demande un volume de données avec deux gigaoctets d'espace utilisable. L'utilisateur n'a pas besoin de connaître le site **volumeMode** pour estimer correctement la taille de la demande de volume persistant (PVC) requise. Le CDI choisit automatiquement la combinaison optimale des attributs **accessModes** et **volumeMode**. Ces valeurs optimales sont basées sur le type de stockage ou sur les valeurs par défaut que vous définissez dans votre profil de stockage. Si vous souhaitez fournir des valeurs personnalisées, elles remplacent les valeurs calculées par le système.

Exemple de définition d'un volume de données

```
apiVersion: cdi.kubevirt.io/v1beta1
```

```
kind: DataVolume
metadata:
  name: <datavolume> 1
spec:
  source:
    pvc: 2
      namespace: "<source_namespace>" 3
      name: "<my_vm_disk>" 4
  storage: 5
    resources:
      requests:
        storage: 2Gi 6
    storageClassName: <storage_class> 7
```

1. Le nom du nouveau volume de données.
2. Indiquer que la source de l'importation est une revendication de volume persistant (PVC) existante.
3. L'espace de noms dans lequel le PVC source existe.
4. Le nom du PVC source.
5. Indique l'allocation à l'aide de l'API de stockage.
6. Spécifie la quantité d'espace disponible que vous demandez pour le PVC.
7. Facultatif : Le nom de la classe de stockage. Si la classe de stockage n'est pas spécifiée, la classe de stockage par défaut du système est utilisée.

10.22.3.3. Création de volumes de données à l'aide de l'API PVC

Lorsque vous créez un volume de données à l'aide de l'API PVC, l'interface de données conteneurisées (CDI) crée le volume de données en fonction de ce que vous spécifiez pour les champs suivants :

- **accessModes** (**ReadWriteOnce**, **ReadWriteMany**, ou **ReadOnlyMany**)
- **volumeMode** (**Filesystem** ou **Block**)
- **capacity** de **storage** (**5Gi**, par exemple)

Dans le fichier YAML suivant, l'utilisation de l'API PVC alloue un volume de données d'une capacité de stockage de deux gigaoctets. Vous spécifiez un mode d'accès de **ReadWriteMany** pour permettre la migration en direct. Comme vous connaissez les valeurs que votre système peut prendre en charge, vous spécifiez le stockage **Block** au lieu de la valeur par défaut, **Filesystem**.

Exemple de définition d'un volume de données

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <datavolume> 1
spec:
  source:
    pvc: 2
```

```
    namespace: "<source_namespace>" 3
    name: "<my_vm_disk>" 4
  pvc: 5
    accessModes: 6
     - ReadWriteMany
    resources:
      requests:
        storage: 2Gi 7
    volumeMode: Block 8
    storageClassName: <storage_class> 9
```

1. Le nom du nouveau volume de données.
2. Dans la section **source**, **pvc** indique que la source de l'importation est une revendication de volume persistante (PVC) existante.
3. L'espace de noms dans lequel le PVC source existe.
4. Le nom du PVC source.
5. Indique l'allocation à l'aide de l'API PVC.
6. **accessModes** est nécessaire lors de l'utilisation de l'API PVC.
7. Spécifie la quantité d'espace que vous demandez pour votre volume de données.
8. Spécifie que la destination est un PVC de bloc.
9. En option, spécifiez la classe de stockage. Si la classe de stockage n'est pas spécifiée, la classe de stockage par défaut du système est utilisée.

IMPORTANT

Lorsque vous allouez explicitement un volume de données à l'aide de l'API PVC et que vous n'utilisez pas **volumeMode: Block**, tenez compte de la surcharge du système de fichiers.

L'overhead du système de fichiers est la quantité d'espace requise par le système de fichiers pour maintenir ses métadonnées. La quantité d'espace requise pour les métadonnées du système de fichiers dépend du système de fichiers. Si vous ne tenez pas compte des frais généraux du système de fichiers dans votre demande de capacité de stockage, vous risquez d'obtenir une revendication de volume persistant (PVC) sous-jacente qui n'est pas assez grande pour accueillir le disque de votre machine virtuelle.

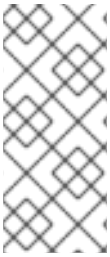
Si vous utilisez l'API de stockage, le CDI tiendra compte des frais généraux du système de fichiers et demandera une demande de volume persistant (PVC) plus importante pour s'assurer que votre demande d'allocation aboutisse.

10.22.3.4. Personnalisation du profil de stockage

Vous pouvez spécifier des paramètres par défaut en modifiant l'objet **StorageProfile** pour la classe de stockage du provisionneur. Ces paramètres par défaut s'appliquent uniquement à la revendication de volume persistant (PVC) s'ils ne sont pas configurés dans l'objet **DataVolume**.

Conditions préalables

- Assurez-vous que la configuration prévue est prise en charge par la classe de stockage et son fournisseur. La spécification d'une configuration incompatible dans un profil de stockage entraîne l'échec du provisionnement de volume.



NOTE

Une section **status** vide dans un profil de stockage indique qu'un stockeur n'est pas reconnu par l'interface de données conteneurisées (CDI). La personnalisation d'un profil de stockage est nécessaire si vous avez un fournisseur de stockage qui n'est pas reconnu par l'interface de données conteneurisées. Dans ce cas, l'administrateur définit les valeurs appropriées dans le profil de stockage pour garantir la réussite des attributions.

AVERTISSEMENT

Si vous créez un volume de données et omettez les attributs YAML et que ces attributs ne sont pas définis dans le profil de stockage, le stockage demandé ne sera pas alloué et la revendication de volume persistant (PVC) sous-jacente ne sera pas créée.

Procédure

1. Modifiez le profil de stockage. Dans cet exemple, le provisionneur n'est pas reconnu par CDI :

```
oc edit -n openshift-cnv storageprofile <storage_class>
```

Exemple de profil de stockage

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: StorageProfile
metadata:
  name: <unknown_provisioner_class>
#   ...
spec: {}
status:
  provisioner: <unknown_provisioner>
  storageClass: <unknown_provisioner_class>
```

2. Fournir les valeurs d'attribut nécessaires dans le profil de stockage :

Exemple de profil de stockage

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: StorageProfile
metadata:
  name: <unknown_provisioner_class>
#   ...
spec:
  claimPropertySets:
```

```
  - accessModes:
    - ReadWriteOnce (1)
    volumeMode:
      Filesystem (2)
status:
  provisioner: <unknown_provisioner>
  storageClass: <unknown_provisioner_class>
```

(1) Le site **accessModes** que vous avez sélectionné.

(2) Le site **volumeMode** que vous avez sélectionné.

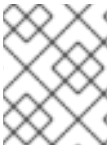
Après avoir enregistré vos modifications, les valeurs sélectionnées apparaissent dans l'élément du profil de stockage **status**.

10.22.3.4.1. Définition d'une stratégie de clonage par défaut à l'aide d'un profil de stockage

Vous pouvez utiliser les profils de stockage pour définir une méthode de clonage par défaut pour une classe de stockage, en créant un site *cloning strategy*. La définition de stratégies de clonage peut s'avérer utile, par exemple, si votre fournisseur de stockage ne prend en charge que certaines méthodes de clonage. Cela vous permet également de sélectionner une méthode qui limite l'utilisation des ressources ou maximise les performances.

Les stratégies de clonage peuvent être spécifiées en définissant l'attribut **cloneStrategy** d'un profil de stockage sur l'une de ces valeurs :

- **snapshot** - Cette méthode est utilisée par défaut lorsque des instantanés sont configurés. Cette stratégie de clonage utilise un instantané de volume temporaire pour cloner le volume. Le provisionneur de stockage doit prendre en charge les instantanés CSI.
- **copy** - Cette méthode utilise un pod source et un pod cible pour copier les données du volume source vers le volume cible. Le clonage assisté par l'hôte est la méthode de clonage la moins efficace.
- **csi-clone** - Cette méthode utilise l'API de clonage CSI pour cloner efficacement un volume existant sans utiliser d'instantané de volume intermédiaire. Contrairement à **snapshot** ou **copy**, qui sont utilisés par défaut si aucun profil de stockage n'est défini, le clonage de volume CSI n'est utilisé que si vous le spécifiez dans l'objet **StorageProfile** pour la classe de stockage du provisionneur.



NOTE

Vous pouvez également définir des stratégies de clonage à l'aide de l'interface de gestion sans modifier la valeur par défaut de **claimPropertySets** dans votre section YAML **spec**.

Exemple de profil de stockage

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: StorageProfile
metadata:
  name: <provisioner_class>
#   ...
spec:
  claimPropertySets:
```

```
  - accessModes:
    - ReadWriteOnce (1)
    volumeMode:
      Filesystem (2)
  cloneStrategy:
    csi-clone (3)
status:
  provisioner: <provisioner>
  storageClass: <provisioner_class>
```

1. Le site **accessModes** que vous avez sélectionné.
2. Le site **volumeMode** que vous avez sélectionné.
3. La méthode de clonage par défaut de votre choix. Dans cet exemple, le clonage du volume CSI est spécifié.

10.22.3.5. Ressources supplémentaires

- Création de classes de stockage
- Remplacer la valeur par défaut de la surcharge du système de fichiers
- Clonage d'un volume de données à l'aide du clonage intelligent

10.22.4. Réservation d'espace PVC pour les frais généraux du système de fichiers

Par défaut, OpenShift Virtualization réserve de l'espace pour les données de surcharge du système de fichiers dans les réclamations de volume persistant (PVC) qui utilisent le mode de volume **Filesystem**. Vous pouvez définir le pourcentage pour réserver de l'espace à cette fin globalement et pour des classes de stockage spécifiques.

10.22.4.1. Comment la surcharge du système de fichiers affecte l'espace pour les disques des machines virtuelles

Lorsque vous ajoutez un disque de machine virtuelle à une revendication de volume persistant (PVC) qui utilise le mode de volume **Filesystem**, vous devez vous assurer qu'il y a suffisamment d'espace sur le PVC pour :

- Le disque de la machine virtuelle.
- L'espace réservé aux frais généraux du système de fichiers, tels que les métadonnées

Par défaut, OpenShift Virtualization réserve 5,5 % de l'espace PVC pour l'overhead, ce qui réduit d'autant l'espace disponible pour les disques des machines virtuelles.

Vous pouvez configurer une autre valeur de frais généraux en modifiant l'objet **HCO**. Vous pouvez modifier la valeur globalement et vous pouvez spécifier des valeurs pour des classes de stockage spécifiques.

10.22.4.2. Remplacer la valeur par défaut de la surcharge du système de fichiers

Modifiez la quantité d'espace PVC (persistent volume claim) que OpenShift Virtualization réserve à l'overhead du système de fichiers en modifiant l'attribut **spec.config.filesystemOverhead** de l'objet **HCO**.

Conditions préalables

- Installez le CLI OpenShift (**oc**).

Procédure

1. Ouvrez l'objet **HCO** pour le modifier en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Modifiez les champs **spec.config.filesystemOverhead** en y ajoutant les valeurs que vous avez choisies :

```
...
spec:
  config:
    filesystemOverhead:
      global: "<new_global_value>" 1
      storageClass:
        <storage_class_name>: "<new_value_for_this_storage_class>" 2
```

1. Le pourcentage de frais généraux du système de fichiers par défaut utilisé pour toutes les classes de stockage qui n'ont pas encore de valeur définie. Par exemple, **global: "0.07"** réserve 7 % du PVC à l'overhead du système de fichiers.

2. Le pourcentage de frais généraux du système de fichiers pour la classe de stockage spécifiée. Par exemple, **mystorageclass: "0.04"** modifie la valeur de surcharge par défaut pour les PVC dans la classe de stockage **mystorageclass** à 4 %.

3. Enregistrez et quittez l'éditeur pour mettre à jour l'objet **HCO**.

Vérification

- Affichez l'état de **CDIConfig** et vérifiez vos modifications en exécutant l'une des commandes suivantes :
  Pour vérifier de manière générale les modifications apportées à **CDIConfig**:

```
$ oc get cdiconfig -o yaml
```

  Pour consulter les modifications apportées à **CDIConfig**:

```
$ oc get cdiconfig -o jsonpath='{.items..status.filesystemOverhead}'
```

10.22.5. Configurer CDI pour travailler avec des espaces de noms ayant un quota de ressources de calcul

Vous pouvez utiliser l'importateur de données conteneurisées (CDI) pour importer, télécharger et cloner des disques de machines virtuelles dans des espaces de noms soumis à des restrictions de ressources de CPU et de mémoire.

10.22.5.1. A propos des quotas de CPU et de mémoire dans un espace de noms

Un *resource quota*, défini par l'objet **ResourceQuota**, impose des restrictions à un espace de noms qui limitent la quantité totale de ressources de calcul pouvant être consommées par les ressources de cet espace de noms.

La ressource personnalisée (CR) **HyperConverged** définit la configuration de l'utilisateur pour l'importateur de données conteneurisées (CDI). Les valeurs de demande et de limite de CPU et de mémoire sont fixées à la valeur par défaut de **0**, ce qui garantit que les pods créés par CDI qui ne spécifient pas de besoins en ressources de calcul reçoivent les valeurs par défaut et sont autorisés à fonctionner dans un espace de noms restreint par un quota.

10.22.5.2. Dépassement des valeurs par défaut de l'unité centrale et de la mémoire

Modifiez les paramètres par défaut pour les demandes et les limites de CPU et de mémoire pour votre cas d'utilisation en ajoutant la strophe **spec.resourceRequirements.storageWorkloads** à la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- Installez le CLI OpenShift (**oc**).

Procédure

1. Modifiez le CR **HyperConverged** en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Ajoutez la strophe **spec.resourceRequirements.storageWorkloads** à la CR, en définissant les valeurs en fonction de votre cas d'utilisation. Par exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  resourceRequirements:
    storageWorkloads:
      limits:
        cpu: "500m"
        memory: "2Gi"
      requests:
        cpu: "250m"
        memory: "1Gi"
```

3. Sauvegardez et quittez l'éditeur pour mettre à jour le CR **HyperConverged**.

10.22.5.3. Ressources supplémentaires

- Quotas de ressources par projet

10.22.6. Gestion des annotations sur les volumes de données

Les annotations sur les volumes de données (DV) permettent de gérer le comportement des pods. Vous pouvez ajouter une ou plusieurs annotations à un volume de données, qui se propage ensuite aux pods d'importation créés.

10.22.6.1. Exemple : Annotations sur les volumes de données

Cet exemple montre comment vous pouvez configurer les annotations de volume de données (DV) pour contrôler le réseau utilisé par le module d'importation. L'annotation **v1.multus-cni.io/default-network: bridge-network** fait en sorte que le module utilise le réseau multus nommé **bridge-network** comme réseau par défaut. Si vous souhaitez que le module importateur utilise à la fois le réseau par défaut du cluster et le réseau multus secondaire, utilisez l'annotation **k8s.v1.cni.cncf.io/networks: <network_name>**.

Exemple d'annotation du réseau Multus

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: dv-ann
  annotations:
    v1.multus-cni.io/default-network: bridge-network 1
spec:
  source:
    http:
      url: "example.exampleurl.com"
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 1Gi
```

1 Annotation du réseau Multus

10.22.7. Utilisation de la pré-allocation pour les volumes de données

L'importateur de données conteneurisées peut pré-allouer de l'espace disque pour améliorer les performances d'écriture lors de la création de volumes de données.

Vous pouvez activer la pré-allocation pour des volumes de données spécifiques.

10.22.7.1. À propos de la pré-allocation

L'importateur de données conteneurisées (CDI) peut utiliser le mode de pré-allocation QEMU pour les volumes de données afin d'améliorer les performances d'écriture. Vous pouvez utiliser le mode de pré-allocation pour les opérations d'importation et de téléchargement et lors de la création de volumes de données vierges.

Si la pré-allocation est activée, CDI utilise la meilleure méthode de pré-allocation en fonction du système de fichiers sous-jacent et du type de périphérique :

fallocate

Si le système de fichiers le prend en charge, CDI utilise l'appel **fallocate** du système d'exploitation pour préallouer l'espace en utilisant la fonction **posix_fallocate**, qui alloue des blocs et les marque comme non initialisés.

full

Si le mode **fallocate** ne peut pas être utilisé, le mode **full** alloue de l'espace pour l'image en écrivant des données dans la mémoire sous-jacente. En fonction de l'emplacement de stockage, tout l'espace vide alloué peut être mis à zéro.

10.22.7.2. Activation de la pré-allocation pour un volume de données

Vous pouvez activer la pré-allocation pour des volumes de données spécifiques en incluant le champ **spec.preallocation** dans le manifeste du volume de données. Vous pouvez activer le mode de pré-allocation dans la console web ou en utilisant le CLI OpenShift (**oc**).

Le mode de pré-affectation est pris en charge pour tous les types de sources CDI.

Procédure

- Spécifiez le champ **spec.preallocation** dans le manifeste du volume de données :

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: preallocated-datavolume
spec:
  source: 1
    ...
  pvc:
    ...
  preallocation: true 2
```

1. Tous les types de sources CDI prennent en charge la pré-allocation, mais celle-ci est ignorée pour les opérations de clonage.
2. Le champ **preallocation** est un booléen dont la valeur par défaut est false.

10.22.8. Téléchargement d'images de disques locaux à l'aide de la console web

Vous pouvez télécharger un fichier image disque stocké localement à l'aide de la console web.

10.22.8.1. Conditions préalables

- Vous devez disposer d'un fichier image de machine virtuelle au format IMG, ISO ou QCOW2.
- Si vous avez besoin d'un espace scratch conformément à la matrice des opérations supportées par le CDI, vous devez d'abord définir une classe de stockage ou préparer l'espace scratch du CDI pour que cette opération se déroule correctement.

10.22.8.2. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.22.8.3. Téléchargement d'un fichier image à l'aide de la console web

Utilisez la console Web pour télécharger un fichier image vers une nouvelle revendication de volume persistant (PVC). Vous pourrez ensuite utiliser ce PVC pour attacher l'image à de nouvelles machines virtuelles.

Conditions préalables

- Vous devez posséder l'un des éléments suivants
  - Un fichier image brut de machine virtuelle au format ISO ou IMG.
  - Un fichier image de machine virtuelle au format QCOW2.
- Pour de meilleurs résultats, comprimez votre fichier image selon les directives suivantes avant de le télécharger :
  - Compresser un fichier image brut en utilisant **xz** ou **gzip**.

NOTE

L'utilisation d'un fichier d'image brute compressé permet le téléchargement le plus efficace.

  - Compresser un fichier image QCOW2 en utilisant la méthode recommandée pour votre client :
    - Si vous utilisez un client Linux, *sparsify* le fichier QCOW2 en utilisant l'outil virt-sparsify.
    - Si vous utilisez un client Windows, comprimez le fichier QCOW2 en utilisant **xz** ou **gzip**.

Procédure

1. Dans le menu latéral de la console web, cliquez sur **Storage → Persistent Volume Claims**.
2. Cliquez sur la liste déroulante **Create Persistent Volume Claim**pour la développer.
3. Cliquez sur **With Data Upload Form**pour ouvrir la page **Upload Data to Persistent Volume Claim**.
4. Cliquez sur **Browse** pour ouvrir le gestionnaire de fichiers et sélectionnez l'image que vous souhaitez télécharger, ou faites glisser le fichier dans le champ **Drag a file here or browse to upload**.
5. Facultatif : Définir cette image comme image par défaut pour un système d'exploitation spécifique.
   a. Cochez la case **Attach this data to a virtual machine operating system**.
   b. Sélectionnez un système d'exploitation dans la liste.
6. Le champ **Persistent Volume Claim Name**est automatiquement rempli avec un nom unique et ne peut pas être modifié. Notez le nom attribué au PVC afin de pouvoir l'identifier ultérieurement, si nécessaire.
7. Sélectionnez une classe de stockage dans la liste **Storage Class**.
8. Dans le champ **Size**, entrez la valeur de la taille du PVC. Sélectionnez l'unité de mesure correspondante dans la liste déroulante.

AVERTISSEMENT

La taille du PVC doit être supérieure à la taille du disque virtuel non compressé.

9. Sélectionnez une adresse **Access Mode** correspondant à la classe de stockage que vous avez sélectionnée.
10. Cliquez sur **Upload**.

10.22.8.4. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.22.9. Téléchargement d'images de disques locaux à l'aide de l'outil virtctl

Vous pouvez télécharger une image disque stockée localement vers un volume de données nouveau ou existant à l'aide de l'utilitaire de ligne de commande **virtctl**.

10.22.9.1. Conditions préalables

- Installer **virtctl**.

- Si vous avez besoin d'un espace scratch conformément à la matrice des opérations supportées par le CDI, vous devez d'abord définir une classe de stockage ou préparer l'espace scratch du CDI pour que cette opération se déroule correctement.

10.22.9.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.22.9.3. Création d'un volume de données à télécharger

Vous pouvez créer manuellement un volume de données avec une source de données **upload** à utiliser pour télécharger des images de disque locales.

Procédure

1. Créez une configuration de volume de données qui spécifie **spec: source: upload{}**:

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <upload-datavolume> 1
spec:
  source:
    upload: {}
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: <2Gi> 2
```

1. Le nom du volume de données.
2. La taille du volume de données. Assurez-vous que cette valeur est supérieure ou égale à la taille du disque que vous téléchargez.

2. Créez le volume de données en exécutant la commande suivante :

```
oc create -f <upload-datavolume>.yaml
```

10.22.9.4. Téléchargement d'une image de disque local vers un volume de données

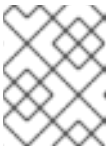
Vous pouvez utiliser l'utilitaire CLI **virtctl** pour télécharger une image de disque locale d'une machine cliente vers un volume de données (DV) dans votre cluster. Vous pouvez utiliser un DV qui existe déjà dans votre cluster ou créer un nouveau DV au cours de cette procédure.

NOTE

Après avoir téléchargé une image de disque local, vous pouvez l'ajouter à une machine virtuelle.

Conditions préalables

- Vous devez posséder l'un des éléments suivants
  - Un fichier image brut de machine virtuelle au format ISO ou IMG.
  - Un fichier image de machine virtuelle au format QCOW2.
- Pour de meilleurs résultats, comprimez votre fichier image selon les directives suivantes avant de le télécharger :
  - Compresser un fichier image brut en utilisant **xz** ou **gzip**.



NOTE

L'utilisation d'un fichier d'image brute compressé permet le téléchargement le plus efficace.

  - Compresser un fichier image QCOW2 en utilisant la méthode recommandée pour votre client :
    - Si vous utilisez un client Linux, *sparsify* le fichier QCOW2 en utilisant l'outil virt-sparsify.
    - Si vous utilisez un client Windows, comprimez le fichier QCOW2 en utilisant **xz** ou **gzip**.
- Le paquet **kubevirt-virtctl** doit être installé sur la machine cliente.
- La machine cliente doit être configurée pour faire confiance au certificat du routeur OpenShift Container Platform.

Procédure

1. Identifiez les éléments suivants :
   - Le nom du volume de données de téléchargement que vous souhaitez utiliser. Si ce volume de données n'existe pas, il est créé automatiquement.
   - La taille du volume de données, si vous souhaitez qu'il soit créé pendant la procédure de téléchargement. La taille doit être supérieure ou égale à la taille de l'image disque.
   - L'emplacement du fichier de l'image du disque de la machine virtuelle que vous souhaitez télécharger.

2. Téléchargez l'image disque en exécutant la commande **virtctl image-upload**. Spécifiez les paramètres que vous avez identifiés à l'étape précédente. Par exemple :

```
$ virtctl image-upload dv <datavolume_name> \ 1
--size=<datavolume_size> \ 2
--image-path=</path/to/image> \ 3
```

1. Le nom du volume de données.
2. La taille du volume de données. Par exemple : **--size=500Mi**, **--size=1G**
3. Le chemin d'accès au fichier de l'image du disque de la machine virtuelle.

NOTE

- Si vous ne souhaitez pas créer un nouveau volume de données, omettez le paramètre **--size** et incluez l'indicateur **--no-create**.
- Lors du téléchargement d'une image de disque vers un PVC, la taille du PVC doit être supérieure à la taille du disque virtuel non compressé.
- Pour autoriser les connexions de serveur non sécurisées lors de l'utilisation de HTTPS, utilisez le paramètre **--insecure**. Sachez que lorsque vous utilisez le paramètre **--insecure**, l'authenticité du point de terminaison du téléchargement est vérifiée par **not**.

3. Facultatif. Pour vérifier qu'un volume de données a été créé, affichez tous les volumes de données en exécutant la commande suivante :

```
$ oc get dvs
```

10.22.9.5. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.22.9.6. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.22.10. Téléchargement d'une image de disque local vers un volume de données de stockage par blocs

Vous pouvez télécharger une image de disque local dans un volume de données en bloc à l'aide de l'utilitaire de ligne de commande **virtctl**.

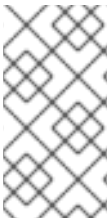
Dans ce flux de travail, vous créez un périphérique bloc local à utiliser comme volume persistant, vous associez ce volume bloc à un volume de données **upload** et vous utilisez **virtctl** pour télécharger l'image du disque local dans le volume de données.

10.22.10.1. Conditions préalables

- Installer **virtctl**.
- Si vous avez besoin d'un espace scratch conformément à la matrice des opérations supportées par le CDI, vous devez d'abord définir une classe de stockage ou préparer l'espace scratch du CDI pour que cette opération se déroule correctement.

10.22.10.2. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.22.10.3. À propos des volumes persistants en bloc

Un volume persistant (PV) en mode bloc est un PV soutenu par un périphérique en mode bloc brut. Ces volumes n'ont pas de système de fichiers et peuvent offrir des avantages en termes de performances pour les machines virtuelles en réduisant les frais généraux.

Les volumes de blocs bruts sont approvisionnés en spécifiant **volumeMode: Block** dans les spécifications PV et PVC (persistent volume claim).

10.22.10.4. Création d'un volume persistant en bloc local

Créez un volume persistant (PV) local en bloc sur un nœud en remplissant un fichier et en le montant en tant que périphérique en boucle. Vous pouvez ensuite référencer ce périphérique en boucle dans un manifeste PV en tant que volume **Block** et l'utiliser comme périphérique de bloc pour une image de machine virtuelle.

Procédure

1. Connectez-vous en tant que **root** au nœud sur lequel vous souhaitez créer le PV local. Cette procédure utilise **node01** pour ses exemples.

2. Créez un fichier et remplissez-le de caractères nuls afin qu'il puisse être utilisé comme périphérique de bloc. L'exemple suivant crée un fichier **loop10** d'une taille de 2 Go (20 blocs de 100 Mo) :

```
$ dd if=/dev/zero of=<loop10> bs=100M count=20
```

3. Monter le fichier **loop10** en tant que périphérique en boucle.

```
$ losetup </dev/loop10>d3 <loop10> 1 2
```

1. Chemin d'accès au fichier où le périphérique loop est monté.
2. Le fichier créé à l'étape précédente doit être monté en tant que périphérique de boucle.

4. Créez un manifeste **PersistentVolume** qui fait référence au périphérique en boucle monté.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: <local-block-pv10>
  annotations:
spec:
  local:
    path: </dev/loop10> 1
  capacity:
    storage: <2Gi>
  volumeMode: Block 2
  storageClassName: local 3
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - <node01> 4
```

1. Chemin d'accès du dispositif de boucle sur le nœud.
2. Indique qu'il s'agit d'un PV en bloc.
3. Facultatif : Définissez une classe de stockage pour le PV. Si vous ne le faites pas, la valeur par défaut du cluster est utilisée.
4. Le nœud sur lequel le périphérique de bloc a été monté.

5. Créer le bloc PV.

```
# oc create -f <local-block-pv10.yaml>1
```

1. Le nom de fichier du volume persistant créé à l'étape précédente.

10.22.10.5. Création d'un volume de données à télécharger

Vous pouvez créer manuellement un volume de données avec une source de données **upload** à utiliser pour télécharger des images de disque locales.

Procédure

1. Créez une configuration de volume de données qui spécifie **spec: source: upload{}**:

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <upload-datavolume> 1
spec:
  source:
    upload: {}
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: <2Gi> 2
```

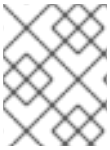
1. Le nom du volume de données.
2. La taille du volume de données. Assurez-vous que cette valeur est supérieure ou égale à la taille du disque que vous téléchargez.

2. Créez le volume de données en exécutant la commande suivante :

```
oc create -f <upload-datavolume>.yaml
```

10.22.10.6. Téléchargement d'une image de disque local vers un volume de données

Vous pouvez utiliser l'utilitaire CLI **virtctl** pour télécharger une image de disque locale d'une machine cliente vers un volume de données (DV) dans votre cluster. Vous pouvez utiliser un DV qui existe déjà dans votre cluster ou créer un nouveau DV au cours de cette procédure.



NOTE

Après avoir téléchargé une image de disque local, vous pouvez l'ajouter à une machine virtuelle.

Conditions préalables

- Vous devez posséder l'un des éléments suivants
  - Un fichier image brut de machine virtuelle au format ISO ou IMG.
  - Un fichier image de machine virtuelle au format QCOW2.
- Pour de meilleurs résultats, comprimez votre fichier image selon les directives suivantes avant de le télécharger :
  - Compresser un fichier image brut en utilisant **xz** ou **gzip**.

NOTE

L'utilisation d'un fichier d'image brute compressé permet le téléchargement le plus efficace.

  - Compresser un fichier image QCOW2 en utilisant la méthode recommandée pour votre client :
    - Si vous utilisez un client Linux, *sparsify* le fichier QCOW2 en utilisant l'outil virt-sparsify.
    - Si vous utilisez un client Windows, comprimez le fichier QCOW2 en utilisant **xz** ou **gzip**.
- Le paquet **kubevirt-virtctl** doit être installé sur la machine cliente.
- La machine cliente doit être configurée pour faire confiance au certificat du routeur OpenShift Container Platform.

Procédure

1. Identifiez les éléments suivants :
   - Le nom du volume de données de téléchargement que vous souhaitez utiliser. Si ce volume de données n'existe pas, il est créé automatiquement.
   - La taille du volume de données, si vous souhaitez qu'il soit créé pendant la procédure de téléchargement. La taille doit être supérieure ou égale à la taille de l'image disque.
   - L'emplacement du fichier de l'image du disque de la machine virtuelle que vous souhaitez télécharger.
2. Téléchargez l'image disque en exécutant la commande **virtctl image-upload**. Spécifiez les paramètres que vous avez identifiés à l'étape précédente. Par exemple :

```
$ virtctl image-upload dv <datavolume_name> \ 1
--size=<datavolume_size> \ 2
--image-path=</path/to/image> \ 3
```

1. Le nom du volume de données.
2. La taille du volume de données. Par exemple : **--size=500Mi**, **--size=1G**
3. Le chemin d'accès au fichier de l'image du disque de la machine virtuelle.

NOTE

- Si vous ne souhaitez pas créer un nouveau volume de données, omettez le paramètre **--size** et incluez l'indicateur **--no-create**.
- Lors du téléchargement d'une image de disque vers un PVC, la taille du PVC doit être supérieure à la taille du disque virtuel non compressé.
- Pour autoriser les connexions de serveur non sécurisées lors de l'utilisation de HTTPS, utilisez le paramètre **--insecure**. Sachez que lorsque vous utilisez le paramètre **--insecure**, l'authenticité du point de terminaison du téléchargement est vérifiée par **not**.

3. Facultatif. Pour vérifier qu'un volume de données a été créé, affichez tous les volumes de données en exécutant la commande suivante :

```
$ oc get dvs
```

10.22.10.7. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.22.10.8. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.22.11. Gestion des instantanés de machines virtuelles

Vous pouvez créer et supprimer des instantanés de machines virtuelles (VM) pour les VM, qu'elles soient hors tension (offline) ou sous tension (online). Vous ne pouvez restaurer qu'une VM hors tension (hors ligne). OpenShift Virtualization prend en charge les instantanés de VM sur les éléments suivants :

- Red Hat OpenShift Data Foundation
- Tout autre fournisseur de stockage dans le cloud avec le pilote Container Storage Interface (CSI) qui prend en charge l'API Kubernetes Volume Snapshot

Les instantanés en ligne ont un délai par défaut de cinq minutes (**5m**) qui peut être modifié si nécessaire.

IMPORTANT

Les instantanés en ligne sont pris en charge pour les machines virtuelles dotées de disques virtuels branchés à chaud. Toutefois, les disques branchés à chaud qui ne figurent pas dans les spécifications de la machine virtuelle ne sont pas inclus dans l'instantané.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela garantit que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané de meilleure qualité est pris. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

10.22.11.1. À propos des instantanés de machines virtuelles

Un site *snapshot* représente l'état et les données d'une machine virtuelle (VM) à un moment précis. Vous pouvez utiliser un instantané pour restaurer une VM existante à un état antérieur (représenté par l'instantané) à des fins de sauvegarde et de reprise après sinistre ou pour revenir rapidement à une version de développement antérieure.

Un instantané de VM est créé à partir d'une VM hors tension (état arrêté) ou sous tension (état en cours d'exécution).

Lorsqu'il prend un instantané d'une VM en cours d'exécution, le contrôleur vérifie que l'agent invité QEMU est installé et en cours d'exécution. Si c'est le cas, il gèle le système de fichiers de la VM avant de prendre l'instantané et dégèle le système de fichiers une fois l'instantané pris.

L'instantané stocke une copie de chaque volume de l'interface de stockage de conteneurs (CSI) attaché à la VM, ainsi qu'une copie de la spécification et des métadonnées de la VM. Les instantanés ne peuvent pas être modifiés après leur création.

Grâce à la fonction d'instantanés de VM, les administrateurs de clusters et les développeurs d'applications peuvent :

- Créer un nouvel instantané
- Liste de tous les instantanés attachés à une VM spécifique
- Restaurer une VM à partir d'un instantané
- Supprimer un instantané de VM existant

10.22.11.1.1. Contrôleur d'instantanés de machines virtuelles et définitions de ressources personnalisées (CRD)

La fonction d'instantané de VM introduit trois nouveaux objets API définis comme CRD pour la gestion des instantanés :

- **VirtualMachineSnapshot**: Représente une demande de l'utilisateur pour créer un instantané. Il contient des informations sur l'état actuel de la VM.
- **VirtualMachineSnapshotContent**: Représente une ressource provisionnée sur le cluster (un snapshot). Il est créé par le contrôleur d'instantanés de VM et contient des références à toutes les ressources nécessaires pour restaurer la VM.
- **VirtualMachineRestore**: Représente une demande de l'utilisateur pour restaurer une VM à partir d'un instantané.

Le contrôleur d'instantanés VM lie un objet **VirtualMachineSnapshotContent** à l'objet **VirtualMachineSnapshot** pour lequel il a été créé, avec une correspondance univoque.

10.22.11.2. Installation de l'agent invité QEMU sur une machine virtuelle Linux

Le site **qemu-guest-agent** est largement disponible et disponible par défaut dans les machines virtuelles Red Hat. Installez l'agent et démarrez le service.

Pour vérifier si l'agent invité QEMU est installé et fonctionne sur votre machine virtuelle (VM), vérifiez que **AgentConnected** figure dans les spécifications de la VM.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

Procédure

1. Accédez à la ligne de commande de la machine virtuelle via l'une des consoles ou via SSH.
2. Installer l'agent invité QEMU sur la machine virtuelle :

```
$ yum install -y qemu-guest-agent
```

3. Assurez-vous que le service est persistant et démarrez-le :

```
$ systemctl enable --now qemu-guest-agent
```

10.22.11.3. Installation de l'agent invité QEMU sur une machine virtuelle Windows

Pour les machines virtuelles Windows, l'agent invité QEMU est inclus dans les pilotes VirtIO. Installer les pilotes sur une installation Windows existante ou nouvelle.

Pour vérifier si l'agent invité QEMU est installé et fonctionne sur votre machine virtuelle (VM), vérifiez que **AgentConnected** figure dans les spécifications de la VM.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

10.22.11.3.1. Installation des pilotes VirtIO sur une machine virtuelle Windows existante

Installer les pilotes VirtIO à partir du lecteur CD SATA connecté à une machine virtuelle Windows existante.

NOTE

Cette procédure utilise une approche générique pour ajouter des pilotes à Windows. La procédure peut différer légèrement d'une version de Windows à l'autre. Consultez la documentation d'installation de votre version de Windows pour connaître les étapes spécifiques de l'installation.

Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.
2. Se connecter à une session utilisateur Windows.
3. Ouvrez **Device Manager** et développez **Other devices** pour répertorier tous les **Unknown device**.
    a. Ouvrez le site **Device Properties** pour identifier l'appareil inconnu. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Properties**.

b. Cliquez sur l'onglet **Details** et sélectionnez **Hardware Ids** dans la liste **Property**.

c. Comparez le site **Value** pour le site **Hardware Ids** avec les pilotes VirtIO pris en charge.

4. Cliquez avec le bouton droit de la souris sur l'appareil et sélectionnez **Update Driver Software**.

5. Cliquer sur **Browse my computer for driver software**et naviguer jusqu'au lecteur de CD SATA connecté, où se trouvent les pilotes VirtIO. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.

6. Cliquez sur **Next** pour installer le pilote.

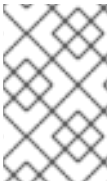
7. Répéter ce processus pour tous les pilotes VirtIO nécessaires.

8. Après l'installation du pilote, cliquez sur **Close** pour fermer la fenêtre.

9. Redémarrez la machine virtuelle pour terminer l'installation du pilote.

10.22.11.3.2. Installation des pilotes VirtIO pendant l'installation de Windows

Installer les pilotes VirtIO à partir du pilote du CD SATA pendant l'installation de Windows.



NOTE

Cette procédure utilise une approche générique de l'installation de Windows et la méthode d'installation peut différer d'une version à l'autre de Windows. Consultez la documentation de la version de Windows que vous installez.

Procédure

1. Démarrez la machine virtuelle et connectez-vous à une console graphique.

2. Commencez le processus d'installation de Windows.

3. Sélectionnez l'installation **Advanced**.

4. La destination de stockage ne sera pas reconnue tant que le pilote n'aura pas été chargé. Cliquez sur **Load driver**.

5. Les pilotes sont attachés à un lecteur de CD SATA. Cliquez sur **OK** et recherchez dans le lecteur de CD le pilote de stockage à charger. Les pilotes sont classés hiérarchiquement en fonction de leur type, du système d'exploitation et de l'architecture du processeur.

6. Répétez les deux étapes précédentes pour tous les pilotes nécessaires.

7. Terminer l'installation de Windows.

10.22.11.4. Créer un instantané de machine virtuelle dans la console web

Vous pouvez créer un instantané de machine virtuelle (VM) à l'aide de la console web.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

L'instantané de la VM ne comprend que les disques qui répondent aux exigences suivantes :

- Il doit s'agir d'un volume de données ou d'une revendication de volume persistant
- Appartenir à une classe de stockage qui prend en charge les instantanés de volume de l'interface de stockage de conteneurs (CSI)

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Si la machine virtuelle est en cours d'exécution, cliquez sur **Actions → Stop** pour l'éteindre.
4. Cliquez sur l'onglet **Snapshots** puis sur **Take Snapshot**.
5. Remplir les champs **Snapshot Name** et **Description** (facultatif).
6. Développez **Disks included in this Snapshot**pour voir les volumes de stockage à inclure dans l'instantané.
7. Si votre VM possède des disques qui ne peuvent pas être inclus dans l'instantané et que vous souhaitez quand même continuer, cochez la case **I am aware of this warning and wish to proceed**.
8. Cliquez sur **Save**.

10.22.11.5. Créer un instantané de machine virtuelle dans le CLI

Vous pouvez créer un instantané de machine virtuelle (VM) pour une VM hors ligne ou en ligne en créant un objet **VirtualMachineSnapshot**. Kubevirt se coordonnera avec l'agent invité QEMU pour créer un instantané de la VM en ligne.

NOTE

Pour créer des instantanés d'une VM en ligne (en cours d'exécution) avec la plus grande intégrité, installez l'agent invité QEMU.

L'agent invité QEMU prend un instantané cohérent en essayant de mettre le système de fichiers de la VM en veille autant que possible, en fonction de la charge de travail du système. Cela permet de s'assurer que les E/S en vol sont écrites sur le disque avant que l'instantané ne soit pris. Si l'agent invité n'est pas présent, la mise en veille n'est pas possible et un instantané est pris au mieux. Les conditions dans lesquelles l'instantané a été pris sont reflétées dans les indications d'instantané qui sont affichées dans la console Web ou dans l'interface de ligne de commande.

Conditions préalables

- Assurez-vous que les réclamations de volumes persistants (PVC) se trouvent dans une classe de stockage qui prend en charge les instantanés de volumes de l'interface de stockage de conteneurs (CSI).
- Installez le CLI OpenShift (**oc**).
- Facultatif : Mettez hors tension la machine virtuelle pour laquelle vous souhaitez créer un instantané.

Procédure

1. Créer un fichier YAML pour définir un objet **VirtualMachineSnapshot** qui spécifie le nom du nouveau **VirtualMachineSnapshot** et le nom de la VM source.
   Par exemple :

```
apiVersion: snapshot.kubevirt.io/v1alpha1
kind: VirtualMachineSnapshot
metadata:
  name: my-vmsnapshot (1)
spec:
  source:
    apiGroup: kubevirt.io
    kind: VirtualMachine
    name: my-vm (2)
```

   (1) Le nom du nouvel objet **VirtualMachineSnapshot**.

   (2) Le nom de la VM source.

2. Créer la ressource **VirtualMachineSnapshot**. Le contrôleur d'instantanés crée un objet **VirtualMachineSnapshotContent**, le lie à **VirtualMachineSnapshot** et met à jour les champs **status** et **readyToUse** de l'objet **VirtualMachineSnapshot**.

```
oc create -f <my-vmsnapshot>.yaml
```

3. Facultatif : si vous prenez un instantané en ligne, vous pouvez utiliser la commande **wait** et surveiller l'état de l'instantané :

   a. Entrez la commande suivante :

```
$ oc wait my-vm my-vmsnapshot --for condition=Ready
```

b. Vérifier l'état de l'instantané :

- **InProgress** - L'opération d'instantané en ligne est toujours en cours.
- **Succeeded** - L'opération d'instantané en ligne s'est terminée avec succès.
- **Failed** - L'opération d'instantané en ligne a échoué.

NOTE

Les instantanés en ligne ont un délai par défaut de cinq minutes (**5m**). Si l'instantané ne se termine pas avec succès dans les cinq minutes, l'état est défini sur **failed**. Par la suite, le système de fichiers sera décongelé et la VM dégelée, mais l'état restera **failed** jusqu'à ce que vous supprimiez l'image de l'instantané qui a échoué.

Pour modifier le délai par défaut, ajoutez l'attribut **FailureDeadline** à la spécification d'instantané VM avec le délai en minutes (**m**) ou en secondes (**s**) que vous souhaitez spécifier avant que l'opération d'instantané ne se termine.

Pour ne pas fixer de délai, vous pouvez spécifier **0**, bien que cela ne soit généralement pas recommandé, car cela peut entraîner une absence de réponse de la part de la VM.

Si vous ne spécifiez pas d'unité de temps telle que **m** ou **s**, la valeur par défaut est la seconde (**s**).

Vérification

1. Vérifiez que l'objet **VirtualMachineSnapshot** est créé et lié à **VirtualMachineSnapshotContent**. L'indicateur **readyToUse** doit être défini sur **true**.

```
$ oc describe vmsnapshot <my-vmsnapshot>
```

Exemple de sortie

```
apiVersion: snapshot.kubevirt.io/v1alpha1
kind: VirtualMachineSnapshot
metadata:
  creationTimestamp: "2020-09-30T14:41:51Z"
  finalizers:
  - snapshot.kubevirt.io/vmsnapshot-protection
  generation: 5
  name: mysnap
  namespace: default
  resourceVersion: "3897"
  selfLink:
/apis/snapshot.kubevirt.io/v1alpha1/namespaces/default/virtualmachinesnapshots/my-
vmsnapshot
  uid: 28eedf08-5d6a-42c1-969c-2eda58e2a78d
spec:
  source:
```

```
    apiGroup: kubevirt.io
    kind: VirtualMachine
    name: my-vm
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2020-09-30T14:42:03Z"
    reason: Operation complete
    status: "False" 1
    type: Progressing
  - lastProbeTime: null
    lastTransitionTime: "2020-09-30T14:42:03Z"
    reason: Operation complete
    status: "True" 2
    type: Ready
  creationTime: "2020-09-30T14:42:03Z"
  readyToUse: true 3
  sourceUID: 355897f3-73a0-4ec4-83d3-3c2df9486f4f
  virtualMachineSnapshotContentName: vmsnapshot-content-28eedf08-5d6a-42c1-969c-
2eda58e2a78d 4
```

1. Le champ **status** de la condition **Progressing** indique si l'instantané est toujours en cours de création.
2. Le champ **status** de la condition **Ready** indique si le processus de création de l'instantané est terminé.
3. Indique si l'instantané est prêt à être utilisé.
4. Spécifie que l'instantané est lié à un objet **VirtualMachineSnapshotContent** créé par le contrôleur d'instantanés.

2. Vérifiez la propriété **spec:volumeBackups** de la ressource **VirtualMachineSnapshotContent** pour vérifier que les PVC prévus sont inclus dans l'instantané.

10.22.11.6. Vérification de la création d'un cliché en ligne à l'aide d'indications de cliché

Les indications de snapshot sont des informations contextuelles sur les opérations de snapshot de machine virtuelle (VM) en ligne. Les indications ne sont pas disponibles pour les opérations de snapshot de machine virtuelle (VM) hors ligne. Les indications sont utiles pour décrire les détails de la création d'un instantané en ligne.

Conditions préalables

- Pour afficher les indications, vous devez avoir tenté de créer un instantané de VM en ligne à l'aide de l'interface CLI ou de la console Web.

Procédure

1. Affichez la sortie des indications de l'instantané en effectuant l'une des opérations suivantes :
   - Pour les instantanés créés à l'aide de l'interface de programmation, affichez l'indicateur dans l'objet YAML **VirtualMachineSnapshot**, dans le champ **status**.

- Pour les instantanés créés à l'aide de la console Web, cliquez sur **VirtualMachineSnapshot > Status** dans l'écran **Snapshot details**.

2. Vérifiez l'état de l'instantané de votre VM en ligne :

   - **Online** indique que la machine virtuelle était en cours d'exécution lors de la création de l'instantané en ligne.

   - **NoGuestAgent** indique que l'agent invité QEMU n'était pas en cours d'exécution lors de la création d'un instantané en ligne. L'agent invité QEMU n'a pas pu être utilisé pour geler et dégeler le système de fichiers, soit parce que l'agent invité QEMU n'était pas installé ou en cours d'exécution, soit en raison d'une autre erreur.

10.22.11.7. Restauration d'une machine virtuelle à partir d'un instantané dans la console web

Vous pouvez restaurer une machine virtuelle (VM) à une configuration antérieure représentée par un instantané dans la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Si la machine virtuelle est en cours d'exécution, cliquez sur **Actions → Stop** pour l'éteindre.
4. Cliquez sur l'onglet **Snapshots**. La page affiche une liste des instantanés associés à la machine virtuelle.
5. Choisissez l'une des méthodes suivantes pour restaurer un instantané de VM :
   a. Pour l'instantané que vous souhaitez utiliser comme source pour restaurer la VM, cliquez sur **Restore**.
   b. Sélectionnez un instantané pour ouvrir l'écran **Snapshot Details** et cliquez sur **Actions → Restore VirtualMachineSnapshot**.
6. Dans la fenêtre de confirmation, cliquez sur **Restore** pour restaurer la configuration précédente de la VM, représentée par l'instantané.

10.22.11.8. Restauration d'une machine virtuelle à partir d'un instantané dans le CLI

Vous pouvez restaurer une machine virtuelle (VM) existante dans une configuration antérieure à l'aide d'un instantané de VM. Vous ne pouvez restaurer qu'à partir d'un instantané de VM hors ligne.

Conditions préalables

- Installez le CLI OpenShift (**oc**).
- Mettez hors tension la VM que vous souhaitez restaurer dans un état antérieur.

Procédure

1. Créez un fichier YAML pour définir un objet **VirtualMachineRestore** qui spécifie le nom de la VM à restaurer et le nom de l'instantané à utiliser comme source.
   Par exemple :

```
apiVersion: snapshot.kubevirt.io/v1alpha1
kind: VirtualMachineRestore
metadata:
  name: my-vmrestore 1
spec:
  target:
    apiGroup: kubevirt.io
    kind: VirtualMachine
    name: my-vm 2
  virtualMachineSnapshotName: my-vmsnapshot 3
```

1. Le nom du nouvel objet **VirtualMachineRestore**.
2. Le nom de la VM cible que vous souhaitez restaurer.
3. Le nom de l'objet **VirtualMachineSnapshot** à utiliser comme source.

2. Créez la ressource **VirtualMachineRestore**. Le contrôleur d'instantanés met à jour les champs d'état de l'objet **VirtualMachineRestore** et remplace la configuration existante de la VM par le contenu de l'instantané.

```
$ oc create -f <my-vmrestore>.yaml
```

Vérification

- Vérifiez que la VM est restaurée dans l'état précédent représenté par l'instantané. L'indicateur **complete** doit être défini sur **true**.

```
oc get vmrestore <my-vmrestore>
```

Exemple de sortie

```
apiVersion: snapshot.kubevirt.io/v1alpha1
kind: VirtualMachineRestore
metadata:
creationTimestamp: "2020-09-30T14:46:27Z"
generation: 5
name: my-vmrestore
namespace: default
ownerReferences:
- apiVersion: kubevirt.io/v1
  blockOwnerDeletion: true
  controller: true
  kind: VirtualMachine
  name: my-vm
  uid: 355897f3-73a0-4ec4-83d3-3c2df9486f4f
  resourceVersion: "5512"
  selfLink:
/apis/snapshot.kubevirt.io/v1alpha1/namespaces/default/virtualmachinerestores/my-
vmrestore
  uid: 71c679a8-136e-46b0-b9b5-f57175a6a041
  spec:
    target:
```

```
    apiGroup: kubevirt.io
    kind: VirtualMachine
    name: my-vm
  virtualMachineSnapshotName: my-vmsnapshot
  status:
  complete: true (1)
  conditions:
  - lastProbeTime: null
  lastTransitionTime: "2020-09-30T14:46:28Z"
  reason: Operation complete
  status: "False" (2)
  type: Progressing
  - lastProbeTime: null
  lastTransitionTime: "2020-09-30T14:46:28Z"
  reason: Operation complete
  status: "True" (3)
  type: Ready
  deletedDataVolumes:
  - test-dv1
  restoreTime: "2020-09-30T14:46:28Z"
  restores:
  - dataVolumeName: restore-71c679a8-136e-46b0-b9b5-f57175a6a041-datavolumedisk1
  persistentVolumeClaim: restore-71c679a8-136e-46b0-b9b5-f57175a6a041-
datavolumedisk1
  volumeName: datavolumedisk1
  volumeSnapshotName: vmsnapshot-28eedf08-5d6a-42c1-969c-2eda58e2a78d-volume-
datavolumedisk1
```

(1) Indique si le processus de restauration de la VM dans l'état représenté par l'instantané est terminé.

(2) Le champ **status** de la condition **Progressing** précise si la VM est toujours en cours de restauration.

(3) Le champ **status** de la condition **Ready** indique si le processus de restauration de la VM est terminé.

10.22.11.9. Suppression d'un snapshot de machine virtuelle dans la console web

Vous pouvez supprimer un snapshot de machine virtuelle existant en utilisant la console web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Snapshots**. La page affiche une liste des instantanés associés à la machine virtuelle.
4. Cliquez sur le menu Options ⋮ de l'instantané de la machine virtuelle que vous souhaitez supprimer et sélectionnez **Delete VirtualMachineSnapshot**.

5. Dans la fenêtre de confirmation, cliquez sur **Delete** pour supprimer l'instantané.

10.22.11.10. Suppression d'un snapshot de machine virtuelle dans le CLI

Vous pouvez supprimer un instantané de machine virtuelle (VM) existant en supprimant l'objet **VirtualMachineSnapshot** approprié.

Conditions préalables

- Installez le CLI OpenShift (**oc**).

Procédure

- Supprimer l'objet **VirtualMachineSnapshot**. Le contrôleur d'instantanés supprime l'objet **VirtualMachineSnapshot** ainsi que l'objet **VirtualMachineSnapshotContent** qui lui est associé.

```
oc delete vmsnapshot <my-vmsnapshot>
```

Vérification

- Vérifiez que l'instantané est supprimé et qu'il n'est plus attaché à cette VM :

```
$ oc get vmsnapshot
```

10.22.11.11. Ressources supplémentaires

- Instantanés de volumes CSI

10.22.12. Déplacement d'un disque de machine virtuelle locale vers un autre nœud

Les machines virtuelles qui utilisent un volume de stockage local peuvent être déplacées de manière à s'exécuter sur un nœud spécifique.

Il se peut que vous souhaitiez déplacer la machine virtuelle vers un nœud spécifique pour les raisons suivantes :

- Le nœud actuel a des limites à la configuration du stockage local.
- Le nouveau nœud est mieux optimisé pour la charge de travail de cette machine virtuelle.

Pour déplacer une machine virtuelle qui utilise le stockage local, vous devez cloner le volume sous-jacent en utilisant un volume de données. Une fois l'opération de clonage terminée, vous pouvez modifier la configuration de la machine virtuelle afin qu'elle utilise le nouveau volume de données ou ajouter le nouveau volume de données à une autre machine virtuelle.

ASTUCE

Lorsque vous activez la pré-allocation globalement ou pour un seul volume de données, l'importateur de données conteneurisées (CDI) pré-allocation l'espace disque pendant le clonage. La pré-allocation améliore les performances d'écriture. Pour plus d'informations, voir Utilisation de la pré-allocation pour les volumes de données.

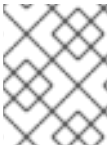
NOTE

Les utilisateurs qui n'ont pas le rôle **cluster-admin** ont besoin d' autorisations supplémentaires pour cloner des volumes dans des espaces de noms.

10.22.12.1. Clonage d'un volume local vers un autre nœud

Vous pouvez déplacer un disque de machine virtuelle pour qu'il s'exécute sur un nœud spécifique en clonant la revendication de volume persistant (PVC) sous-jacente.

Pour vous assurer que le disque de la machine virtuelle est cloné sur le nœud correct, vous devez créer un nouveau volume persistant (PV) ou en identifier un sur le nœud correct. Appliquez une étiquette unique au volume persistant afin qu'il puisse être référencé par le volume de données.



NOTE

Le PV de destination doit être de la même taille ou plus grand que le PVC source. Si le PV de destination est plus petit que le PVC source, l'opération de clonage échoue.

Conditions préalables

- La machine virtuelle ne doit pas être en cours d'exécution. Mettez la machine virtuelle hors tension avant de cloner le disque de la machine virtuelle.

Procédure

1. Créer un nouveau PV local sur le nœud ou identifier un PV local déjà présent sur le nœud :
   - Créer un PV local qui inclut les paramètres **nodeAffinity.nodeSelectorTerms**. Le manifeste suivant crée un PV local **10Gi** sur **node01**.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: <destination-pv> 1
  annotations:
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 10Gi 2
  local:
    path: /mnt/local-storage/local/disk1 3
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - node01 4
  persistentVolumeReclaimPolicy: Delete
  storageClassName: local
  volumeMode: Filesystem
```

1. Le nom du PV.
2. La taille du PV. Vous devez allouer suffisamment d'espace, sinon l'opération de clonage échoue. La taille doit être égale ou supérieure à celle du PVC source.
3. Le chemin de montage sur le nœud.
4. Le nom du nœud où vous voulez créer le PV.

- Identifier un PV qui existe déjà sur le nœud cible. Vous pouvez identifier le nœud où un PV est provisionné en consultant le champ **nodeAffinity** dans sa configuration :

```
$ oc get pv <destination-pv> -o yaml
```

L'extrait suivant montre que le PV est sur **node01**:

Exemple de sortie

```
...
spec:
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname 1
          operator: In
          values:
          - node01 2
...
```

1. La clé **kubernetes.io/hostname** utilise le nom d'hôte du nœud pour sélectionner un nœud.
2. Le nom d'hôte du nœud.

2. Ajouter une étiquette unique au PV :

```
oc label pv <destination-pv> node=node01
```

3. Créez un manifeste de volume de données qui fait référence aux éléments suivants :

- Le nom PVC et l'espace de noms de la machine virtuelle.
- L'étiquette que vous avez appliquée au PV à l'étape précédente.
- Taille du PV de destination.

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <clone-datavolume> 1
spec:
  source:
```

```
  pvc:
    name: "<source-vm-disk>" 2
    namespace: "<source-namespace>" 3
pvc:
  accessModes:
    - ReadWriteOnce
  selector:
    matchLabels:
      node: node01 4
  resources:
    requests:
      storage: <10Gi> 5
```

1. Le nom du nouveau volume de données.
2. Le nom du PVC source. Si vous ne connaissez pas le nom du PVC, vous pouvez le trouver dans la configuration de la machine virtuelle : **spec.volumes.persistentVolumeClaim.claimName**.
3. L'espace de noms dans lequel le PVC source existe.
4. L'étiquette que vous avez appliquée au PV à l'étape précédente.
5. Taille du PV de destination.

4. Commencez l'opération de clonage en appliquant le manifeste du volume de données à votre cluster :

```
oc apply -f <clone-datavolume.yaml>
```

Le volume de données clone le PVC de la machine virtuelle dans le PV sur le nœud spécifique.

10.22.13. Extension du stockage virtuel par l'ajout d'images de disques vierges

Vous pouvez augmenter votre capacité de stockage ou créer de nouvelles partitions de données en ajoutant des images de disques vierges à OpenShift Virtualization.

10.22.13.1. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus

nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.22.13.2. Création d'une image de disque vierge avec des volumes de données

Vous pouvez créer une nouvelle image de disque vierge dans une revendication de volume persistant en personnalisant et en déployant un fichier de configuration de volume de données.

Conditions préalables

- Au moins un volume persistant disponible.
- Installez le CLI OpenShift (**oc**).

Procédure

1. Modifiez le manifeste **DataVolume**:

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: blank-image-datavolume
spec:
  source:
    blank: {}
  pvc:
    # Optional: Set the storage class or omit to accept the default
    # storageClassName: "hostpath"
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 500Mi
```

2. Créez l'image de disque vierge en exécutant la commande suivante :

```
oc create -f <blank-image-datavolume>.yaml
```

10.22.13.3. Ressources supplémentaires

- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.

10.22.14. Clonage d'un volume de données à l'aide de smart-cloning

Le clonage intelligent est une fonctionnalité intégrée de Red Hat OpenShift Data Foundation. Le clonage intelligent est plus rapide et plus efficace que le clonage assisté par l'hôte.

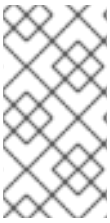
Vous n'avez aucune action à effectuer pour activer le clonage intelligent, mais vous devez vous assurer que votre environnement de stockage est compatible avec le clonage intelligent pour utiliser cette fonctionnalité.

Lorsque vous créez un volume de données avec une source de revendication de volume persistant

(PVC), vous lancez automatiquement le processus de clonage. Vous recevez toujours un clone du volume de données, que votre environnement prenne ou non en charge le clonage intelligent. Cependant, vous ne bénéficierez des avantages en termes de performances du clonage intelligent que si votre fournisseur de stockage prend en charge le clonage intelligent.

10.22.14.1. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).



NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

10.22.14.2. À propos du clonage intelligent

Lorsqu'un volume de données est cloné de manière intelligente, il se produit ce qui suit :

1. Un instantané de la revendication de volume persistant (PVC) source est créé.
2. Un PVC est créé à partir de l'instantané.
3. L'instantané est supprimé.

10.22.14.3. Clonage d'un volume de données

Conditions préalables

Pour que le clonage intelligent se produise, les conditions suivantes doivent être remplies :

- Votre fournisseur de stockage doit prendre en charge les instantanés.
- Les PVC source et cible doivent être définis pour la même classe de stockage.
- Les PVC source et cible partagent le même **volumeMode**.
- L'objet **VolumeSnapshotClass** doit faire référence à la classe de stockage définie pour les PVC source et cible.

Procédure

Pour lancer le clonage d'un volume de données :

1. Créez un fichier YAML pour un objet **DataVolume** qui spécifie le nom du nouveau volume de données ainsi que le nom et l'espace de noms du PVC source. Dans cet exemple, comme vous

spécifiez l'API **storage**, il n'est pas nécessaire de spécifier **accessModes** ou **volumeMode**. Les valeurs optimales seront calculées automatiquement pour vous.

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: <cloner-datavolume> 1
spec:
  source:
    pvc:
      namespace: "<source-namespace>" 2
      name: "<my-favorite-vm-disk>" 3
  storage: 4
    resources:
      requests:
        storage: <2Gi> 5
```

1. Le nom du nouveau volume de données.
2. L'espace de noms dans lequel le PVC source existe.
3. Le nom du PVC source.
4. Spécifie l'allocation avec l'API **storage**
5. Taille du nouveau volume de données.

2. Commencez à cloner le PVC en créant le volume de données :

```
oc create -f <cloner-datavolume>.yaml
```

NOTE

Les volumes de données empêchent le démarrage d'une machine virtuelle avant que le PVC ne soit préparé. Vous pouvez donc créer une machine virtuelle qui fait référence au nouveau volume de données pendant que le PVC se clone.

10.22.14.4. Ressources supplémentaires

- Clonage de la revendication de volume persistant d'un disque de machine virtuelle dans un nouveau volume de données
- Configurez le mode de pré-affectation pour améliorer les performances d'écriture pour les opérations sur les volumes de données.
- Personnalisation du profil de stockage

10.22.15. Création et utilisation des sources de démarrage

Une source d'amorçage contient un système d'exploitation (OS) amorçable et tous les paramètres de configuration de l'OS, tels que les pilotes.

Vous utilisez une source de démarrage pour créer des modèles de machines virtuelles avec des configurations spécifiques. Ces modèles peuvent être utilisés pour créer un nombre quelconque de machines virtuelles disponibles.

Des visites de démarrage rapide sont disponibles dans la console web d'OpenShift Container Platform pour vous aider à créer une source de démarrage personnalisée, à télécharger une source de démarrage et à effectuer d'autres tâches. Sélectionnez **Quick Starts** dans le menu **Help** pour afficher les visites guidées de démarrage rapide.

10.22.15.1. À propos des machines virtuelles et des sources de démarrage

Les machines virtuelles se composent d'une définition de machine virtuelle et d'un ou plusieurs disques qui sont sauvegardés par des volumes de données. Les modèles de machines virtuelles vous permettent de créer des machines virtuelles à l'aide de spécifications prédéfinies.

Chaque modèle de machine virtuelle nécessite une source de démarrage, qui est une image de disque de machine virtuelle entièrement configurée, y compris les pilotes configurés. Chaque modèle de machine virtuelle contient une définition de machine virtuelle avec un pointeur vers la source de démarrage. Chaque source de démarrage a un nom et un espace de noms prédéfinis. Pour certains systèmes d'exploitation, une source de démarrage est automatiquement fournie. Si elle n'est pas fournie, l'administrateur doit préparer une source d'amorçage personnalisée.

Les sources d'amorçage fournies sont automatiquement mises à jour avec la dernière version du système d'exploitation. Pour les sources d'amorçage mises à jour automatiquement, les réclamations de volumes persistants (PVC) sont créées avec la classe de stockage par défaut de la grappe. Si vous sélectionnez une autre classe de stockage par défaut après la configuration, vous devez supprimer les volumes de données existants dans l'espace de noms du cluster qui sont configurés avec la classe de stockage par défaut précédente.

Pour utiliser la fonctionnalité des sources de démarrage, installez la dernière version d'OpenShift Virtualization. L'espace de noms **openshift-virtualization-os-images** active la fonctionnalité et est installé avec l'opérateur OpenShift Virtualization. Une fois la fonctionnalité de source de démarrage installée, vous pouvez créer des sources de démarrage, les attacher à des modèles et créer des machines virtuelles à partir des modèles.

Définir une source de démarrage à l'aide d'une revendication de volume persistant (PVC) qui est remplie en téléchargeant un fichier local, en clonant un PVC existant, en l'important à partir d'un registre ou par URL. Attachez une source de démarrage à un modèle de machine virtuelle à l'aide de la console Web. Une fois que la source de démarrage est attachée à un modèle de machine virtuelle, vous créez un nombre quelconque de machines virtuelles prêtes à l'emploi et entièrement configurées à partir du modèle.

10.22.15.2. Importation d'une image RHEL comme source de démarrage

Vous pouvez importer une image Red Hat Enterprise Linux (RHEL) comme source de démarrage en spécifiant une URL pour l'image.

Conditions préalables

- Vous devez avoir accès à une page web contenant l'image du système d'exploitation. Par exemple : Télécharger la page web de Red Hat Enterprise Linux avec les images.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Templates** dans le menu latéral.

2. Identifiez le modèle RHEL pour lequel vous souhaitez configurer une source de démarrage et cliquez sur **Add source**.

3. Dans la fenêtre **Add boot source to template**, sélectionnez **URL (creates PVC)** dans la liste **Boot source type**.

4. Cliquez sur **RHEL download page** pour accéder au portail client de Red Hat. Une liste des installateurs et des images disponibles est affichée sur la page Télécharger Red Hat Enterprise Linux.

5. Identifiez l'image de l'invité Red Hat Enterprise Linux KVM que vous souhaitez télécharger. Cliquez avec le bouton droit de la souris sur **Download Now** et copiez l'URL de l'image.

6. Dans la fenêtre **Add boot source to template**, collez l'URL dans le champ **Import URL** et cliquez sur **Save and import**.

Vérification

1. Vérifiez que le modèle affiche une coche verte dans la colonne **Boot source** de la page **Templates**.

Vous pouvez maintenant utiliser ce modèle pour créer des machines virtuelles RHEL.

10.22.15.3. Ajout d'une source de démarrage pour un modèle de machine virtuelle

Une source de démarrage peut être configurée pour tout modèle de machine virtuelle que vous souhaitez utiliser pour créer des machines virtuelles ou des modèles personnalisés. Lorsque les modèles de machines virtuelles sont configurés avec une source de démarrage, ils sont étiquetés **Source available** sur la page **Templates**. Après avoir ajouté une source de démarrage à un modèle, vous pouvez créer une nouvelle machine virtuelle à partir du modèle.

Il existe quatre méthodes pour sélectionner et ajouter une source de démarrage dans la console web :

- **Upload local file (creates PVC)**
- **URL (creates PVC)**
- **Clone (creates PVC)**
- **Registry (creates PVC)**

Conditions préalables

- Pour ajouter une source de démarrage, vous devez être connecté en tant qu'utilisateur avec le rôle RBAC **os-images.kubevirt.io:edit** ou en tant qu'administrateur. Vous n'avez pas besoin de privilèges spéciaux pour créer une machine virtuelle à partir d'un modèle auquel une source de démarrage a été ajoutée.
- Pour télécharger un fichier local, le fichier image du système d'exploitation doit exister sur votre machine locale.
- Pour importer via une URL, il faut avoir accès au serveur web contenant l'image du système d'exploitation. Par exemple : la page web de Red Hat Enterprise Linux avec les images.
- Pour cloner un PVC existant, l'accès au projet avec un PVC est nécessaire.

- Pour importer via le registre, il faut avoir accès au registre des conteneurs.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Cliquez sur le menu d'options à côté d'un modèle et sélectionnez **Edit boot source**.
3. Cliquez sur **Add disk**.
4. Dans la fenêtre **Add disk**, sélectionnez **Use this disk as a boot source**
5. Saisissez le nom du disque et sélectionnez une adresse **Source**, par exemple, **Blank (creates PVC)** ou **Use an existing PVC**.
6. Entrez une valeur pour **Persistent Volume Claim size** afin de spécifier la taille du PVC adéquate pour l'image non compressée et tout espace supplémentaire requis.
7. Sélectionnez une adresse **Type**, par exemple **Disk** ou **CD-ROM**.
8. Facultatif : Cliquez sur **Storage class** et sélectionnez la classe de stockage utilisée pour créer le disque. En général, cette classe de stockage est la classe de stockage par défaut qui est créée pour être utilisée par tous les PVC.

NOTE

Les sources d'amorçage fournies sont automatiquement mises à jour avec la dernière version du système d'exploitation. Pour les sources d'amorçage mises à jour automatiquement, les réclamations de volumes persistants (PVC) sont créées avec la classe de stockage par défaut de la grappe. Si vous sélectionnez une autre classe de stockage par défaut après la configuration, vous devez supprimer les volumes de données existants dans l'espace de noms du cluster qui sont configurés avec la classe de stockage par défaut précédente.

9. Facultatif : Clear **Apply optimized StorageProfile settings** pour modifier le mode d'accès ou le mode de volume.
10. Sélectionnez la méthode appropriée pour enregistrer votre source de démarrage :
    a. Cliquez sur **Save and upload** si vous avez téléchargé un fichier local.
    b. Cliquez sur **Save and import** si vous avez importé du contenu à partir d'une URL ou du registre.
    c. Cliquez sur **Save and clone** si vous avez cloné un PVC existant.

Votre modèle de machine virtuelle personnalisé avec une source de démarrage est répertorié sur la page **Catalog**. Vous pouvez utiliser ce modèle pour créer une machine virtuelle.

10.22.15.4. Création d'une machine virtuelle à partir d'un modèle avec une source de démarrage attachée

Après avoir ajouté une source de démarrage à un modèle, vous pouvez créer une machine virtuelle à partir du modèle.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Virtualization → Catalog** dans le menu latéral.
2. Sélectionnez le modèle mis à jour et cliquez sur **Quick create VirtualMachine**.

Le site **VirtualMachine details** s'affiche avec le statut **Starting**.

10.22.15.5. Ressources supplémentaires

- Création de modèles de machines virtuelles
- Importation et mise à jour automatiques de sources de démarrage prédéfinies

10.22.16. Branchement à chaud de disques virtuels

Vous pouvez ajouter ou supprimer des disques virtuels sans arrêter votre machine virtuelle (VM) ou votre instance de machine virtuelle (VMI).

10.22.16.1. À propos de l'enfichage à chaud de disques virtuels

Lorsque vous *hot plug* un disque virtuel, vous attachez un disque virtuel à une instance de machine virtuelle pendant que la machine virtuelle est en cours d'exécution.

Lorsque vous *hot unplug* un disque virtuel, vous détachez un disque virtuel d'une instance de machine virtuelle pendant que la machine virtuelle est en cours d'exécution.

Seuls les volumes de données et les réclamations de volumes persistants (PVC) peuvent être branchés et débranchés à chaud. Il n'est pas possible de brancher ou de débrancher à chaud des disques conteneurs.

Lorsque vous branchez un disque virtuel à chaud, il reste attaché jusqu'à ce que vous le détachiez, même si vous redémarrez la machine virtuelle.

10.22.16.2. À propos de virtio-scsi

Dans OpenShift Virtualization, chaque machine virtuelle (VM) dispose d'un contrôleur **virtio-scsi** afin que les disques branchés à chaud puissent utiliser un bus **scsi**. Le contrôleur **virtio-scsi** surmonte les limites de **virtio** tout en conservant ses avantages en termes de performances. Il est très évolutif et prend en charge le branchement à chaud de plus de 4 millions de disques.

Le site **virtio** n'est pas disponible pour les disques enfichés à chaud car il n'est pas évolutif : chaque disque **virtio** utilise l'un des emplacements PCI Express (PCIe) limités de la VM. Les emplacements PCIe sont également utilisés par d'autres périphériques et doivent être réservés à l'avance ; il se peut donc que les emplacements ne soient pas disponibles à la demande.

10.22.16.3. Branchement à chaud d'un disque virtuel à l'aide de la CLI

Branchez à chaud les disques virtuels que vous souhaitez attacher à une instance de machine virtuelle (VMI) lorsqu'une machine virtuelle est en cours d'exécution.

Conditions préalables

- Vous devez avoir une machine virtuelle en cours d'exécution pour brancher à chaud un disque virtuel.
- Vous devez disposer d'au moins un volume de données ou d'une revendication de volume persistant (PVC) pour le branchement à chaud.

Procédure

- Branchez à chaud un disque virtuel en exécutant la commande suivante :

```
$ virtctl addvolume <virtual-machine|virtual-machine-instance> --volume-name=
<datavolume|PVC> \
[--persist] [--serial=<label-name>]
```

  - Utilisez l'option **--persist** pour ajouter le disque branché à chaud à la spécification de la machine virtuelle en tant que disque virtuel monté de façon permanente. Arrêtez, redémarrez ou réinitialisez la machine virtuelle pour monter le disque virtuel de manière permanente. Après avoir spécifié l'indicateur **--persist**, vous ne pouvez plus brancher ou débrancher à chaud le disque virtuel. L'indicateur **--persist** s'applique aux machines virtuelles et non aux instances de machines virtuelles.
  - L'option **--serial** vous permet d'ajouter une étiquette alphanumérique de votre choix. Cela vous aide à identifier le disque branché à chaud dans une machine virtuelle invitée. Si vous ne spécifiez pas cette option, l'étiquette prend par défaut le nom du volume de données ou du PVC branché à chaud.

10.22.16.4. Débranchement à chaud d'un disque virtuel à l'aide de la CLI

Débranchez à chaud les disques virtuels que vous souhaitez détacher d'une instance de machine virtuelle (VMI) pendant qu'une machine virtuelle est en cours d'exécution.

Conditions préalables

- Votre machine virtuelle doit être en cours d'exécution.
- Vous devez avoir au moins un volume de données ou une revendication de volume persistant (PVC) disponible et branché à chaud.

Procédure

- Débranchez à chaud un disque virtuel en exécutant la commande suivante :

```
virtctl removevolume <virtual-machine|virtual-machine-instance> --volume-name=
<datavolume|PVC>
```

10.22.16.5. Branchement à chaud d'un disque virtuel à l'aide de la console web

Branchez à chaud les disques virtuels que vous souhaitez attacher à une instance de machine virtuelle (VMI) lorsqu'une machine virtuelle est en cours d'exécution. Lorsque vous branchez un disque virtuel à chaud, il reste attaché à la VMI jusqu'à ce que vous le débranchiez.

Conditions préalables

- Vous devez avoir une machine virtuelle en cours d'exécution pour brancher à chaud un disque virtuel.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez la machine virtuelle en cours d'exécution sur laquelle vous souhaitez connecter à chaud un disque virtuel.
3. Sur la page **VirtualMachine details**, cliquez sur l'onglet **Disks**.
4. Cliquez sur **Add disk**.
5. Dans la fenêtre **Add disk (hot plugged)**, remplissez les informations relatives au disque virtuel que vous souhaitez connecter à chaud.
6. Cliquez sur **Save**.

10.22.16.6. Débrancher à chaud un disque virtuel à l'aide de la console web

Débranchez à chaud les disques virtuels que vous souhaitez détacher d'une instance de machine virtuelle (VMI) pendant qu'une machine virtuelle est en cours d'exécution.

Conditions préalables

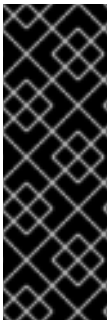
- Votre machine virtuelle doit fonctionner avec un disque connecté à chaud.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez la machine virtuelle en cours d'exécution avec le disque que vous voulez débrancher à chaud pour ouvrir la page **VirtualMachine details**.
3. Dans l'onglet **Disks**, cliquez sur le menu Options  du disque virtuel que vous souhaitez débrancher à chaud.
4. Cliquez sur **Detach**.

10.22.17. Utilisation de disques de conteneurs avec des machines virtuelles

Vous pouvez créer une image de machine virtuelle dans un disque de conteneur et la stocker dans votre registre de conteneurs. Vous pouvez ensuite importer le disque conteneur dans le stockage persistant d'une machine virtuelle ou l'attacher directement à la machine virtuelle pour un stockage éphémère.



IMPORTANT

Si vous utilisez des disques de conteneur de grande taille, le trafic d'E/S risque d'augmenter, ce qui aura un impact sur les nœuds de travail. Cela peut entraîner l'indisponibilité des nœuds. Vous pouvez résoudre ce problème en procédant comme suit

- Élagage des objets **DeploymentConfig**
- Configuration du ramassage des ordures

10.22.17.1. À propos des disques conteneurs

Un disque de conteneur est une image de machine virtuelle stockée en tant qu'image de conteneur dans un registre d'images de conteneur. Vous pouvez utiliser les disques conteneurs pour fournir les mêmes images de disque à plusieurs machines virtuelles et pour créer un grand nombre de clones de machines virtuelles.

Un disque conteneur peut être importé dans une revendication de volume persistant (PVC) en utilisant un volume de données attaché à une machine virtuelle, ou attaché directement à une machine virtuelle en tant que volume éphémère **containerDisk**.

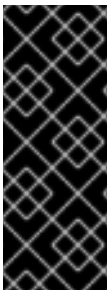
10.22.17.1.1. Importation d'un disque conteneur dans un PVC à l'aide d'un volume de données

Utilisez l'importateur de données conteneurisées (CDI) pour importer le disque conteneur dans un PVC à l'aide d'un volume de données. Vous pouvez ensuite attacher le volume de données à une machine virtuelle pour un stockage persistant.

10.22.17.1.2. Attacher un disque conteneur à une machine virtuelle en tant que volume containerDisk

Un volume **containerDisk** est éphémère. Il est supprimé lorsque la machine virtuelle est arrêtée, redémarrée ou supprimée. Lorsqu'une machine virtuelle avec un volume **containerDisk** démarre, l'image du conteneur est extraite du registre et hébergée sur le nœud qui héberge la machine virtuelle.

Utilisez les volumes **containerDisk** pour les systèmes de fichiers en lecture seule tels que les CD-ROM ou pour les machines virtuelles jetables.



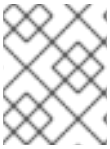
IMPORTANT

L'utilisation des volumes **containerDisk** pour les systèmes de fichiers en lecture-écriture n'est pas recommandée car les données sont temporairement écrites sur le stockage local du nœud d'hébergement. Cela ralentit la migration en direct de la machine virtuelle, par exemple en cas de maintenance du nœud, car les données doivent être migrées vers le nœud de destination. En outre, toutes les données sont perdues en cas de panne de courant ou d'arrêt inattendu du nœud.

10.22.17.2. Préparation d'un disque de conteneur pour les machines virtuelles

Vous devez construire un disque de conteneur avec une image de machine virtuelle et le pousser vers un registre de conteneur avant de pouvoir l'utiliser avec une machine virtuelle. Vous pouvez ensuite importer le disque de conteneur dans un PVC à l'aide d'un volume de données et l'attacher à une machine virtuelle, ou vous pouvez attacher le disque de conteneur directement à une machine virtuelle en tant que volume éphémère **containerDisk**.

La taille d'une image de disque à l'intérieur d'un disque conteneur est limitée par la taille maximale de la couche du registre dans lequel le disque conteneur est hébergé.



NOTE

Pour Red Hat Quay, vous pouvez modifier la taille maximale des couches en éditant le fichier de configuration YAML qui est créé lors du premier déploiement de Red Hat Quay.

Conditions préalables

- Installez **podman** s'il n'est pas déjà installé.
- L'image de la machine virtuelle doit être au format QCOW2 ou RAW.

Procédure

1. Créer un fichier Docker pour construire l'image de la machine virtuelle dans une image de conteneur. L'image de la machine virtuelle doit appartenir à QEMU, qui a un UID de **107**, et être placée dans le répertoire **/disk/** à l'intérieur du conteneur. Les autorisations pour le répertoire **/disk/** doivent ensuite être définies sur **0440**.
   L'exemple suivant utilise l'image de base universelle (UBI) de Red Hat pour gérer ces changements de configuration dans la première étape, et utilise l'image minimale **scratch** dans la deuxième étape pour stocker le résultat :

```
$ cat > Dockerfile << EOF
FROM registry.access.redhat.com/ubi8/ubi:latest AS builder
ADD --chown=107:107 <vm_image>.qcow2 /disk/ 1
RUN chmod 0440 /disk/*

FROM scratch
COPY --from=builder /disk/* /disk/
EOF
```

   1 Où <vm_image> est l'image de la machine virtuelle au format QCOW2 ou RAW.
   Pour utiliser une image de machine virtuelle distante, remplacez **<vm_image>.qcow2** par l'url complète de l'image distante.

2. Construire et étiqueter le conteneur :

```
$ podman build -t <registry>/<container_disk_name>:latest .
```

3. Insérer l'image du conteneur dans le registre :

```
$ podman push <registry>/<container_disk_name>:latest
```

Si votre registre de conteneurs ne dispose pas de TLS, vous devez l'ajouter en tant que registre non sécurisé avant de pouvoir importer des disques de conteneurs dans le stockage persistant.

10.22.17.3. Désactivation de TLS pour un registre de conteneur à utiliser comme registre non sécurisé

Vous pouvez désactiver TLS (transport layer security) pour un ou plusieurs registres de conteneurs en modifiant le champ **insecureRegistries** de la ressource personnalisée **HyperConverged**.

Conditions préalables

- Connectez-vous au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

- Modifiez la ressource personnalisée **HyperConverged** et ajoutez une liste de registres non sécurisés dans le champ **spec.storageImport.insecureRegistries**.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  storageImport:
    insecureRegistries: 1
      - "private-registry-example-1:5000"
      - "private-registry-example-2:5000"
```

1 Remplacez les exemples de cette liste par des noms d'hôtes de registre valides.

10.22.17.4. Prochaines étapes

- Importer le disque conteneur dans le stockage persistant d'une machine virtuelle .
- Créez une machine virtuelle qui utilise un volume **containerDisk** pour le stockage éphémère.

10.22.18. Préparation de l'espace mémoire du CDI

10.22.18.1. A propos des volumes de données

DataVolume sont des ressources personnalisées fournies par le projet Containerized Data Importer (CDI). Les volumes de données orchestrent les opérations d'importation, de clonage et de téléchargement qui sont associées à une revendication de volume persistant (PVC) sous-jacente. Vous pouvez créer un volume de données en tant que ressource autonome ou en utilisant le champ **dataVolumeTemplate** dans la spécification de la machine virtuelle (VM).

NOTE

- Les PVC de disques VM préparés à l'aide de volumes de données autonomes ont un cycle de vie indépendant de celui de la VM. Si vous utilisez le champ **dataVolumeTemplate** dans la spécification de la VM pour préparer le PVC, le PVC partage le même cycle de vie que la VM.

Une fois qu'un PVC est rempli, le volume de données que vous avez utilisé pour créer le PVC n'est plus nécessaire. OpenShift Virtualization active par défaut le ramassage automatique des volumes de données terminés. Les volumes de données autonomes et les volumes de données créés à l'aide de la ressource **dataVolumeTemplate** sont automatiquement mis au rebut une fois terminés.

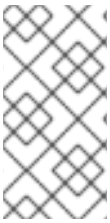
10.22.18.2. À propos de l'espace réservé aux griffes

L'importateur de données conteneurisées (CDI) a besoin d'un espace d'effacement (stockage

temporaire) pour effectuer certaines opérations, telles que l'importation et le téléchargement d'images de machines virtuelles. Au cours de ce processus, CDI fournit un PVC d'espace d'effacement égal à la taille du PVC sauvegardant le volume de données de destination (DV). Le PVC d'espace d'effacement est supprimé une fois l'opération terminée ou interrompue.

Vous pouvez définir la classe de stockage utilisée pour lier l'espace de stockage PVC dans le champ **spec.scratchSpaceStorageClass** de la ressource personnalisée **HyperConverged**.

Si la classe de stockage définie ne correspond pas à une classe de stockage dans le cluster, la classe de stockage par défaut définie pour le cluster est utilisée. Si aucune classe de stockage par défaut n'est définie dans le cluster, la classe de stockage utilisée pour provisionner le DV ou le PVC d'origine est utilisée.



NOTE

CDI exige de demander un espace scratch avec un mode de volume **file**, quel que soit le PVC soutenant le volume de données d'origine. Si le PVC d'origine est soutenu par le mode de volume **block**, vous devez définir une classe de stockage capable de provisionner des PVC en mode de volume **file**.

Approvisionnement manuel
S'il n'y a pas de classes de stockage, CDI utilise tous les PVC du projet qui correspondent aux exigences de taille de l'image. Si aucun PVC ne correspond à ces exigences, le module d'importation CDI reste dans l'état **Pending** jusqu'à ce qu'un PVC approprié soit disponible ou jusqu'à ce qu'une fonction d'expiration du délai d'attente mette fin au module.

10.22.18.3. Les opérations du CDI qui nécessitent un espace de rayures

Type	Raison
Importations du registre	CDI doit télécharger l'image dans un espace de stockage et extraire les couches pour trouver le fichier image. Le fichier image est ensuite transmis à QEMU-IMG pour être converti en disque brut.
Télécharger l'image	QEMU-IMG n'accepte pas d'entrée depuis STDIN. Au lieu de cela, l'image à télécharger est sauvegardée dans l'espace scratch avant d'être transmise à QEMU-IMG pour la conversion.
Importation HTTP d'images archivées	QEMU-IMG ne sait pas comment gérer les formats d'archive pris en charge par CDI. Au lieu de cela, l'image est désarchivée et sauvegardée dans l'espace de stockage avant d'être transmise à QEMU-IMG.
Importations HTTP d'images authentifiées	QEMU-IMG ne gère pas correctement l'authentification. Au lieu de cela, l'image est sauvegardée dans l'espace de stockage et authentifiée avant d'être transmise à QEMU-IMG.

Type	Raison
Importation de certificats personnalisés par HTTP	QEMU-IMG ne gère pas correctement les certificats personnalisés des points de terminaison HTTPS. Au lieu de cela, CDI télécharge l'image dans l'espace de stockage avant de transmettre le fichier à QEMU-IMG.

10.22.18.4. Définition d'une classe de stockage

Vous pouvez définir la classe de stockage que l'importateur de données conteneurisées (CDI) utilise lors de l'allocation de l'espace de stockage en ajoutant le champ **spec.scratchSpaceStorageClass** à la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- Installez le CLI OpenShift (**oc**).

Procédure

1. Modifiez le CR **HyperConverged** en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Ajoutez le champ **spec.scratchSpaceStorageClass** à la CR, en définissant la valeur sur le nom d'une classe de stockage qui existe dans le cluster :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  scratchSpaceStorageClass: "<storage_class>" 1
```

1 Si vous ne spécifiez pas de classe de stockage, CDI utilise la classe de stockage de la revendication de volume persistant qui est en train d'être remplie.

3. Sauvegardez et quittez votre éditeur par défaut pour mettre à jour le CR **HyperConverged**.

10.22.18.5. Matrice des opérations soutenues par le CDI

Cette matrice montre les opérations CDI prises en charge pour les types de contenu par rapport aux points de terminaison, et lesquelles de ces opérations nécessitent de l'espace pour les rayures.

Types de contenu	HTTP	HTTPS	Authentification de base HTTP	Registre	Télécharger
KubeVirt (QCOW2)	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2** ✓ GZ* ✓ XZ*	✓ QCOW2 ✓ GZ* ✓ XZ*	✓ QCOW2* ☐ GZ ☐ XZ	✓ QCOW2* ✓ GZ* ✓ XZ*
KubeVirt (RAW)	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW ✓ GZ ✓ XZ	✓ RAW* ☐ GZ ☐ XZ	✓ RAW* ✓ GZ* ✓ XZ*

✓ Opération supportée

☐ Opération non supportée

* Nécessite de l'espace pour les rayures

** Nécessite de l'espace disque si une autorité de certification personnalisée est requise

10.22.18.6. Ressources supplémentaires

- Provisionnement dynamique

10.22.19. Réutilisation des volumes persistants

Pour réutiliser un volume persistant (PV) provisionné statiquement, vous devez d'abord récupérer le volume. Cela implique la suppression du PV afin que la configuration de stockage puisse être réutilisée.

10.22.19.1. À propos de la récupération des volumes persistants provisionnés statiquement

Lorsque vous récupérez un volume persistant (PV), vous le dissociez d'une revendication de volume persistant (PVC) et vous le supprimez. En fonction du stockage sous-jacent, il se peut que vous deviez supprimer manuellement le stockage partagé.

Vous pouvez ensuite réutiliser la configuration du PV pour créer un PV portant un nom différent.

Les PV provisionnés statiquement doivent avoir une politique de récupération de **Retain** pour être récupérés. Si ce n'est pas le cas, le PV entre dans un état d'échec lorsque le PVC est délié du PV.

IMPORTANT

La politique de récupération de **Recycle** est obsolète dans OpenShift Container Platform 4.

10.22.19.2. Récupération des volumes persistants provisionnés statiquement

Récupérez un volume persistant (PV) provisionné statiquement en déliant la réclamation de volume persistant (PVC) et en supprimant le PV. Il se peut que vous deviez également supprimer manuellement le stockage partagé.

La récupération d'un PV provisionné statiquement dépend du stockage sous-jacent. Cette procédure propose une approche générale qu'il peut être nécessaire de personnaliser en fonction de votre système de stockage.

Procédure

1. Assurez-vous que la politique de récupération du PV est réglée sur **Retain**:

   a. Vérifier la politique de récupération du PV :

   ```
   oc get pv <pv_name> -o yaml | grep 'persistentVolumeReclaimPolicy' $ oc get pv <pv_name> -o yaml
   ```

   b. Si **persistentVolumeReclaimPolicy** n'est pas défini sur **Retain**, modifiez la politique de récupération à l'aide de la commande suivante :

   ```
   $ oc patch pv <pv_name> -p '{"spec":{"persistentVolumeReclaimPolicy":"Retain"}"
   ```

2. S'assurer qu'aucune ressource n'utilise le PV :

   ```
   oc describe pvc <pvc_name> | grep 'Mounted By:' (Monté par :)
   ```

   Supprimez toutes les ressources qui utilisent le PVC avant de continuer.

3. Supprimer le PVC pour libérer le PV :

   ```
   oc delete pvc <pvc_name> $ oc delete pvc <pvc_name>
   ```

4. Facultatif : Exporter la configuration PV dans un fichier YAML. Si vous supprimez manuellement le stockage partagé plus tard dans cette procédure, vous pourrez vous référer à cette configuration. Vous pouvez également utiliser les paramètres **spec** de ce fichier comme base pour créer un nouveau PV avec la même configuration de stockage après avoir récupéré le PV :

   ```
   $ oc get pv <nom_du_v> -o yaml > <nom_du_fichier>.yaml
   ```

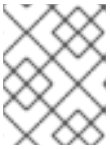
5. Supprimer le PV :

   ```
   oc delete pv <pv_name> $ oc delete pv <pv_name>
   ```

6. Facultatif : Selon le type de stockage, il peut être nécessaire de supprimer le contenu du dossier de stockage partagé :

   ```
   $ rm -rf <path_to_share_storage>
   ```

7. Facultatif : Créez un PV qui utilise la même configuration de stockage que le PV supprimé. Si vous avez exporté la configuration du PV récupéré précédemment, vous pouvez utiliser les paramètres **spec** de ce fichier comme base d'un nouveau manifeste PV :

   NOTE

   Pour éviter tout conflit, il est conseillé de donner au nouvel objet PV un nom différent de celui que vous avez supprimé.

```
oc create -f <nouveau_nom_pv>.yaml
```

Ressources supplémentaires

- Configuration du stockage local pour les machines virtuelles
- La documentation OpenShift Container Platform Storage contient plus d'informations sur le stockage persistant.

10.22.20. Extension du disque d'une machine virtuelle

Vous pouvez augmenter la taille du disque d'une machine virtuelle (VM) pour fournir une plus grande capacité de stockage en redimensionnant la revendication de volume persistant (PVC) du disque.

Cependant, vous ne pouvez pas réduire la taille d'un disque VM.

10.22.20.1. Agrandir le disque d'une machine virtuelle

L'agrandissement du disque de la VM met de l'espace supplémentaire à la disposition de la machine virtuelle. Toutefois, il incombe au propriétaire de la machine virtuelle de décider comment utiliser l'espace de stockage.

Si le disque est un PVC **Filesystem**, le fichier correspondant s'étend à la taille restante tout en réservant de l'espace pour les frais généraux du système de fichiers.

Procédure

1. Modifiez le manifeste **PersistentVolumeClaim** du disque VM que vous souhaitez développer :

```
$ oc edit pvc <nom_du_pvc>
```

2. Modifier la valeur de l'attribut **spec.resource.requests.storage** pour obtenir une taille plus grande.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: vm-disk-expand
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 3Gi 1
...
```

1 Taille du disque de la VM pouvant être augmentée

10.22.20.2. Ressources supplémentaires

- Extension d'un volume de base dans Windows .

- Extension d'une partition de système de fichiers existante sans destruction de données dans Red Hat Enterprise Linux.
- Extension d'un volume logique et de son système de fichiers en ligne dans Red Hat Enterprise Linux.

CHAPITRE 11. MODÈLES DE MACHINES VIRTUELLES

11.1. CRÉATION DE MODÈLES DE MACHINES VIRTUELLES

11.1.1. À propos des modèles de machines virtuelles

Les modèles préconfigurés de machines virtuelles Red Hat sont répertoriés dans la page **Virtualization → Templates**. Ces modèles sont disponibles pour différentes versions de Red Hat Enterprise Linux, Fedora, Microsoft Windows 10 et Microsoft Windows Servers. Chaque modèle de machine virtuelle Red Hat est préconfiguré avec l'image du système d'exploitation, les paramètres par défaut du système d'exploitation, la saveur (CPU et mémoire) et le type de charge de travail (serveur).

La page **Templates** affiche quatre types de modèles de machines virtuelles :

- **Red Hat Supported** sont entièrement pris en charge par Red Hat.
- **User Supported** sont des modèles **Red Hat Supported** qui ont été clonés et créés par les utilisateurs.
- **Red Hat Provided** bénéficient d'une assistance limitée de la part de Red Hat.
- **User Provided** sont des modèles **Red Hat Provided** qui ont été clonés et créés par les utilisateurs.

Vous pouvez utiliser les filtres du modèle **Catalog** pour trier les modèles en fonction d'attributs tels que la disponibilité de la source de démarrage, le système d'exploitation et la charge de travail.

Vous ne pouvez pas modifier ou supprimer un modèle **Red Hat Supported** ou **Red Hat Provided**. Vous pouvez cloner le modèle, l'enregistrer en tant que modèle de machine virtuelle personnalisé, puis le modifier.

Vous pouvez également créer un modèle de machine virtuelle personnalisé en modifiant un exemple de fichier YAML.

11.1.2. À propos des machines virtuelles et des sources de démarrage

Les machines virtuelles se composent d'une définition de machine virtuelle et d'un ou plusieurs disques qui sont sauvegardés par des volumes de données. Les modèles de machines virtuelles vous permettent de créer des machines virtuelles à l'aide de spécifications prédéfinies.

Chaque modèle de machine virtuelle nécessite une source de démarrage, qui est une image de disque de machine virtuelle entièrement configurée, y compris les pilotes configurés. Chaque modèle de machine virtuelle contient une définition de machine virtuelle avec un pointeur vers la source de démarrage. Chaque source de démarrage a un nom et un espace de noms prédéfinis. Pour certains systèmes d'exploitation, une source de démarrage est automatiquement fournie. Si elle n'est pas fournie, l'administrateur doit préparer une source d'amorçage personnalisée.

Les sources d'amorçage fournies sont automatiquement mises à jour avec la dernière version du système d'exploitation. Pour les sources d'amorçage mises à jour automatiquement, les réclamations de volumes persistants (PVC) sont créées avec la classe de stockage par défaut de la grappe. Si vous sélectionnez une autre classe de stockage par défaut après la configuration, vous devez supprimer les volumes de données existants dans l'espace de noms du cluster qui sont configurés avec la classe de stockage par défaut précédente.

Pour utiliser la fonctionnalité des sources de démarrage, installez la dernière version d'OpenShift

Virtualization. L'espace de noms **openshift-virtualization-os-images** active la fonctionnalité et est installé avec l'opérateur OpenShift Virtualization. Une fois la fonctionnalité de source de démarrage installée, vous pouvez créer des sources de démarrage, les attacher à des modèles et créer des machines virtuelles à partir des modèles.

Définir une source de démarrage à l'aide d'une revendication de volume persistant (PVC) qui est remplie en téléchargeant un fichier local, en clonant un PVC existant, en l'important à partir d'un registre ou par URL. Attachez une source de démarrage à un modèle de machine virtuelle à l'aide de la console Web. Une fois que la source de démarrage est attachée à un modèle de machine virtuelle, vous créez un nombre quelconque de machines virtuelles prêtes à l'emploi et entièrement configurées à partir du modèle.

11.1.3. Créer un modèle de machine virtuelle dans la console web

Vous créez un modèle de machine virtuelle en éditant un exemple de fichier YAML dans la console web d'OpenShift Container Platform.

Procédure

1. Dans la console web, cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Optionnel : Utilisez le menu déroulant **Project** pour modifier le projet associé au nouveau modèle. Par défaut, tous les modèles sont enregistrés dans le projet **openshift**.
3. Cliquez sur **Create Template**.
4. Spécifiez les paramètres du modèle en modifiant le fichier YAML.
5. Cliquez sur **Create**.
   Le modèle est affiché sur la page **Templates**.
6. Optionnel : Cliquez sur **Download** pour télécharger et enregistrer le fichier YAML.

11.1.4. Ajout d'une source de démarrage pour un modèle de machine virtuelle

Une source de démarrage peut être configurée pour tout modèle de machine virtuelle que vous souhaitez utiliser pour créer des machines virtuelles ou des modèles personnalisés. Lorsque les modèles de machines virtuelles sont configurés avec une source de démarrage, ils sont étiquetés **Source available** sur la page **Templates**. Après avoir ajouté une source de démarrage à un modèle, vous pouvez créer une nouvelle machine virtuelle à partir du modèle.

Il existe quatre méthodes pour sélectionner et ajouter une source de démarrage dans la console web :

- **Upload local file (creates PVC)**
- **URL (creates PVC)**
- **Clone (creates PVC)**
- **Registry (creates PVC)**

Conditions préalables

- Pour ajouter une source de démarrage, vous devez être connecté en tant qu'utilisateur avec le rôle RBAC **os-images.kubevirt.io:edit** ou en tant qu'administrateur. Vous n'avez pas besoin de privilèges spéciaux pour créer une machine virtuelle à partir d'un modèle auquel une source de

démarrage a été ajoutée.

- Pour télécharger un fichier local, le fichier image du système d'exploitation doit exister sur votre machine locale.
- Pour importer via une URL, il faut avoir accès au serveur web contenant l'image du système d'exploitation. Par exemple : la page web de Red Hat Enterprise Linux avec les images.
- Pour cloner un PVC existant, l'accès au projet avec un PVC est nécessaire.
- Pour importer via le registre, il faut avoir accès au registre des conteneurs.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Cliquez sur le menu d'options à côté d'un modèle et sélectionnez **Edit boot source**.
3. Cliquez sur **Add disk**.
4. Dans la fenêtre **Add disk**, sélectionnez **Use this disk as a boot source**
5. Saisissez le nom du disque et sélectionnez une adresse **Source**, par exemple, **Blank (creates PVC)** ou **Use an existing PVC**.
6. Entrez une valeur pour **Persistent Volume Claim size** afin de spécifier la taille du PVC adéquate pour l'image non compressée et tout espace supplémentaire requis.
7. Sélectionnez une adresse **Type**, par exemple **Disk** ou **CD-ROM**.
8. Facultatif : Cliquez sur **Storage class** et sélectionnez la classe de stockage utilisée pour créer le disque. En général, cette classe de stockage est la classe de stockage par défaut qui est créée pour être utilisée par tous les PVC.

NOTE

Les sources d'amorçage fournies sont automatiquement mises à jour avec la dernière version du système d'exploitation. Pour les sources d'amorçage mises à jour automatiquement, les réclamations de volumes persistants (PVC) sont créées avec la classe de stockage par défaut de la grappe. Si vous sélectionnez une autre classe de stockage par défaut après la configuration, vous devez supprimer les volumes de données existants dans l'espace de noms du cluster qui sont configurés avec la classe de stockage par défaut précédente.

9. Facultatif : Clear **Apply optimized StorageProfile settings** pour modifier le mode d'accès ou le mode de volume.
10. Sélectionnez la méthode appropriée pour enregistrer votre source de démarrage :
    a. Cliquez sur **Save and upload** si vous avez téléchargé un fichier local.
    b. Cliquez sur **Save and import** si vous avez importé du contenu à partir d'une URL ou du registre.
    c. Cliquez sur **Save and clone** si vous avez cloné un PVC existant.

Votre modèle de machine virtuelle personnalisé avec une source de démarrage est répertorié sur la page **Catalog**. Vous pouvez utiliser ce modèle pour créer une machine virtuelle.

11.1.4.1. Champs du modèle de machine virtuelle pour l'ajout d'une source de démarrage

Le tableau suivant décrit les champs de la fenêtre **Add boot source to template**. Cette fenêtre s'affiche lorsque vous cliquez sur **Add source** pour un modèle de machine virtuelle sur la page **Virtualization → Templates**.

Nom	Paramètres	Description
Type de source de démarrage	Télécharger un fichier local (crée un PVC)	Téléchargez un fichier à partir de votre appareil local. Les types de fichiers pris en charge sont gz, xz, tar et qcow2.
	URL (crée un PVC)	Importer le contenu d'une image disponible à partir d'un point de terminaison HTTP ou HTTPS. Obtenez l'URL du lien de téléchargement à partir de la page web où le téléchargement de l'image est disponible et entrez ce lien URL dans le champ **Import URL**. Exemple : Pour une image Red Hat Enterprise Linux, connectez-vous au portail client Red Hat, accédez à la page de téléchargement d'images et copiez l'URL du lien de téléchargement pour l'image d'invité KVM.
	PVC (crée du PVC)	Utiliser un PVC déjà disponible dans le cluster et le cloner.
	Registre (crée le PVC)	Spécifiez le conteneur de système d'exploitation amorçable situé dans un registre et accessible depuis le cluster. Exemple : kubevirt/cirros-registry-dis-demo.
Fournisseur de la source		Champ facultatif. Ajoutez un texte descriptif sur la source du modèle ou le nom de l'utilisateur qui a créé le modèle. Exemple : Red Hat.
Paramètres de stockage avancés	Classe de stockage	La classe de stockage utilisée pour créer le disque.

Nom	Paramètres	Description
	Mode d'accès	Mode d'accès au volume persistant. Les modes d'accès pris en charge sont **Single User (RWO)**, **Shared Access (RWX)**, **Read Only (ROX)**. Si **Single User (RWO)** est sélectionné, le disque peut être monté en lecture/écriture par un seul nœud. Si **Shared Access (RWX)** est sélectionné, le disque peut être monté en lecture/écriture par plusieurs nœuds. La carte de configuration **kubevirt-storage-class-defaults** fournit les modes d'accès par défaut pour les volumes de données. La valeur par défaut est définie en fonction de la meilleure option pour chaque classe de stockage dans le cluster.  **NOTE** L'accès partagé (RWX) est nécessaire pour certaines fonctionnalités, telles que la migration en direct des machines virtuelles entre les nœuds.
	Mode volume	Définit si le volume persistant utilise un système de fichiers formaté ou un état de bloc brut. Les modes pris en charge sont **Block** et **Filesystem**. La carte de configuration **kubevirt-storage-class-defaults** fournit les valeurs par défaut du mode de volume pour les volumes de données. La valeur par défaut est définie en fonction de la meilleure option pour chaque classe de stockage dans le cluster.

11.1.5. Ressources supplémentaires

- Création et utilisation des sources de démarrage
- Personnalisation du profil de stockage

11.2. MODIFIER LES MODÈLES DE MACHINES VIRTUELLES

Vous pouvez modifier un modèle de machine virtuelle dans la console web.

NOTE

Vous ne pouvez pas modifier un modèle fourni par l'Opérateur de virtualisation Red Hat. Si vous clonez le modèle, vous pouvez le modifier.

11.2.1. Modifier un modèle de machine virtuelle dans la console web

Vous pouvez modifier un modèle de machine virtuelle en utilisant la console web d'OpenShift Container Platform ou l'interface de ligne de commande.

La modification d'un modèle de machine virtuelle n'affecte pas les machines virtuelles déjà créées à partir de ce modèle.

Procédure

1. Naviguez vers **Virtualization → Templates** dans la console web.
2. Cliquez sur le À côté d'un modèle de machine virtuelle et sélectionnez l'objet à modifier.
3. Pour modifier un modèle Red Hat, cliquez sur le bouton Options, sélectionnez **Clone** pour créer un modèle personnalisé, puis éditez le modèle personnalisé.

NOTE

Edit boot source reference est désactivée si la source de données du modèle est gérée par la ressource personnalisée **DataImportCron** ou si le modèle n'a pas de référence de volume de données.

4. Cliquez sur **Save**.

11.2.1.1. Ajout d'une interface réseau à un modèle de machine virtuelle

Cette procédure permet d'ajouter une interface réseau à un modèle de machine virtuelle.

Procédure

1. Cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Sélectionnez un modèle de machine virtuelle pour ouvrir l'écran **Template details**.
3. Cliquez sur l'onglet **Network Interfaces**.
4. Cliquez sur **Add Network Interface**.
5. Dans la fenêtre **Add Network Interface**, indiquez les adresses **Name**, **Model**, **Network**, **Type** et **MAC Address** de l'interface réseau.
6. Cliquez sur **Add**.

11.2.1.2. Ajouter un disque virtuel à un modèle de machine virtuelle

Cette procédure permet d'ajouter un disque virtuel à un modèle de machine virtuelle.

Procédure

1. Cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Sélectionnez un modèle de machine virtuelle pour ouvrir l'écran **Template details**.
3. Cliquez sur l'onglet **Disks** puis sur **Add disk**.

4. Dans la fenêtre **Add disk**, indiquez les adresses **Source**, **Name**, **Size**, **Type**, **Interface**, et **Storage Class**.

    a. Facultatif : vous pouvez activer la pré-allocation si vous utilisez une source de disque vierge et que vous avez besoin de performances d'écriture maximales lors de la création de volumes de données. Pour ce faire, cochez la case **Enable preallocation**.

    b. Facultatif : vous pouvez effacer **Apply optimized StorageProfile settings** pour modifier **Volume Mode** et **Access Mode** pour le disque virtuel. Si vous ne spécifiez pas ces paramètres, le système utilise les valeurs par défaut de la carte de configuration **kubevirt-storage-class-defaults**.

5. Cliquez sur **Add**.

11.2.1.3. Édition de CD-ROM pour les modèles

La procédure suivante permet d'éditer des CD-ROM pour les modèles de machines virtuelles.

Procédure

1. Cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Sélectionnez un modèle de machine virtuelle pour ouvrir l'écran **Template details**.
3. Cliquez sur l'onglet **Disks**.
4. Cliquez sur le menu Options  du CD-ROM que vous souhaitez éditer et sélectionnez **Edit**.
5. Dans la fenêtre **Edit CD-ROM**, modifiez les champs : **Source**, **Persistent Volume Claim**, **Name**, **Type**, et **Interface**.
6. Cliquez sur **Save**.

11.3. ACTIVATION DE RESSOURCES DÉDIÉES POUR LES MODÈLES DE MACHINES VIRTUELLES

Les machines virtuelles peuvent disposer de ressources d'un nœud, telles que l'unité centrale, qui leur sont dédiées afin d'améliorer les performances.

11.3.1. À propos des ressources dédiées

Lorsque vous activez les ressources dédiées pour votre machine virtuelle, la charge de travail de votre machine virtuelle est planifiée sur des CPU qui ne seront pas utilisés par d'autres processus. En utilisant des ressources dédiées, vous pouvez améliorer les performances de la machine virtuelle et la précision des prévisions de latence.

11.3.2. Conditions préalables

- Le gestionnaire de CPU doit être configuré sur le nœud. Vérifiez que le nœud porte l'étiquette **cpumanager** = **true** avant de planifier les charges de travail des machines virtuelles.

11.3.3. Activation de ressources dédiées pour un modèle de machine virtuelle

Vous activez les ressources dédiées pour un modèle de machine virtuelle dans l'onglet **Details**. Les machines virtuelles qui ont été créées à partir d'un modèle Red Hat peuvent être configurées avec des ressources dédiées.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Sélectionnez un modèle de machine virtuelle pour ouvrir la page **Template details**.
3. Dans l'onglet **Scheduling**, cliquez sur l'icône représentant un crayon à côté de **Dedicated Resources**.
4. Sélectionnez **Schedule this workload with dedicated resources (guaranteed policy)**
5. Cliquez sur **Save**.

11.4. DÉPLOYER UN MODÈLE DE MACHINE VIRTUELLE DANS UN ESPACE DE NOMS PERSONNALISÉ

Red Hat fournit des modèles de machines virtuelles préconfigurés qui sont installés dans l'espace de noms **openshift**. Le site **ssp-operator** déploie par défaut des modèles de machines virtuelles dans l'espace de noms **openshift**. Les modèles dans l'espace de noms **openshift** sont accessibles à tous les utilisateurs. Ces modèles sont répertoriés sur la page **Virtualization → Templates** pour les différents systèmes d'exploitation.

11.4.1. Création d'un espace de noms personnalisé pour les modèles

Vous pouvez créer un espace de noms personnalisé qui est utilisé pour déployer des modèles de machines virtuelles à utiliser par toute personne ayant des droits d'accès à ces modèles. Pour ajouter des modèles à un espace de noms personnalisé, modifiez la ressource personnalisée (CR) **HyperConverged**, ajoutez **commonTemplatesNamespace** à la spécification et indiquez l'espace de noms personnalisé pour les modèles de machines virtuelles. Une fois la CR **HyperConverged** modifiée, la CR **ssp-operator** remplit les modèles dans l'espace de noms personnalisé.

Conditions préalables

- Install the OpenShift Container Platform CLI **oc**.
- Connectez-vous en tant qu'utilisateur disposant des privilèges d'administrateur de cluster.

Procédure

- Utilisez la commande suivante pour créer votre espace de noms personnalisé :

```
oc create namespace <mycustomnamespace> $ oc create namespace
<mycustomnamespace>
```

11.4.2. Ajouter des modèles à un espace de noms personnalisé

Le site **ssp-operator** déploie par défaut les modèles de machines virtuelles dans l'espace de noms **openshift**. Les modèles dans l'espace de noms **openshift** sont publiquement disponibles pour tous les utilisateurs. Lorsqu'un espace de noms personnalisé est créé et que des modèles sont ajoutés à cet

espace de noms, vous pouvez modifier ou supprimer les modèles de machines virtuelles dans l'espace de noms **openshift**. Pour ajouter des modèles à un espace de noms personnalisé, modifiez la ressource personnalisée (CR) **HyperConverged** qui contient la ressource **ssp-operator**.

Procédure

1. Affichez la liste des modèles de machines virtuelles disponibles dans l'espace de noms **openshift**.

```
$ oc get templates -n openshift
```

2. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

3. Affichez la liste des modèles de machines virtuelles disponibles dans l'espace de noms personnalisé.

```
$ oc get templates -n customnamespace
```

4. Ajoutez l'attribut **commonTemplatesNamespace** et spécifiez l'espace de noms personnalisé. Exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  commonTemplatesNamespace: customnamespace 1
```

1 L'espace de noms personnalisé pour le déploiement des modèles.

5. Enregistrez vos modifications et quittez l'éditeur. Le site **ssp-operator** ajoute les modèles de machines virtuelles qui existent dans l'espace de noms par défaut **openshift** à l'espace de noms personnalisé.

11.4.2.1. Suppression des modèles d'un espace de noms personnalisé

Pour supprimer les modèles de machines virtuelles d'un espace de noms personnalisé, supprimez l'attribut **commonTemplateNamespace** de la ressource personnalisée (CR) **HyperConverged** et supprimez chaque modèle de cet espace de noms personnalisé.

Procédure

1. Modifiez le **HyperConverged** CR dans votre éditeur par défaut en exécutant la commande suivante :

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

2. Supprimer l'attribut **commonTemplateNamespace**.

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  commonTemplatesNamespace: customnamespace (1)
```

(1) L'attribut **commonTemplatesNamespace** à supprimer.

3. Supprimer un modèle spécifique de l'espace de noms personnalisé qui a été supprimé.

```
oc delete templates -n customnamespace <template_name> $ oc delete templates -n
customnamespace <template_name>
```

Vérification

- Vérifiez que le modèle a été supprimé de l'espace de noms personnalisé.

```
$ oc get templates -n customnamespace
```

11.4.2.2. Ressources supplémentaires

- Création de modèles de machines virtuelles

11.5. SUPPRESSION DES MODÈLES DE MACHINES VIRTUELLES

Vous pouvez supprimer des modèles de machine virtuelle personnalisés basés sur des modèles Red Hat en utilisant la console web.

Vous ne pouvez pas supprimer les modèles Red Hat.

11.5.1. Suppression d'un modèle de machine virtuelle dans la console web

La suppression d'un modèle de machine virtuelle le supprime définitivement du cluster.

NOTE

Vous pouvez supprimer des modèles de machine virtuelle personnalisés. Vous ne pouvez pas supprimer les modèles fournis par Red Hat.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → Templates** dans le menu latéral.
2. Cliquez sur le menu Options d'un modèle et sélectionnez **Delete template**.
3. Cliquez sur **Delete**.

CHAPITRE 12. MIGRATION EN DIRECT

12.1. MIGRATION EN DIRECT DE LA MACHINE VIRTUELLE

12.1.1. À propos de la migration en direct

La migration en direct est le processus de déplacement d'une instance de machine virtuelle (IMV) en cours d'exécution vers un autre nœud du cluster sans interrompre la charge de travail virtuelle ou l'accès. Si une IMV utilise la stratégie d'éviction **LiveMigrate**, elle migre automatiquement lorsque le nœud sur lequel elle s'exécute est placé en mode maintenance. Vous pouvez également lancer manuellement une migration en direct en sélectionnant une IMV à migrer.

Vous pouvez utiliser la migration en direct si les conditions suivantes sont remplies :

- Stockage partagé avec mode d'accès **ReadWriteMany** (RWX).
- Mémoire vive et bande passante suffisantes.
- Si la machine virtuelle utilise un modèle de CPU hôte, les nœuds doivent prendre en charge le modèle de CPU hôte de la machine virtuelle.

Par défaut, le trafic de migration en direct est crypté à l'aide du protocole TLS (Transport Layer Security).

12.1.2. Ressources supplémentaires

- Migration d'une instance de machine virtuelle vers un autre nœud
- Suivi de la migration en direct
- Limitation de la migration en direct
- Personnalisation du profil de stockage

12.2. LIMITES ET DÉLAIS DE MIGRATION EN DIRECT

Appliquez des limites et des délais de migration en direct afin que les processus de migration ne submergent pas le cluster. Configurez ces paramètres en modifiant la ressource personnalisée (CR) **HyperConverged**.

12.2.1. Configuration des limites et des délais de migration en direct

Configurez les limites et les délais de migration en direct pour le cluster en mettant à jour la ressource personnalisée (CR) **HyperConverged**, qui se trouve dans l'espace de noms **openshift-cnv**.

Procédure

- Modifiez la CR **HyperConverged** et ajoutez les paramètres de migration en direct nécessaires.

```
$ oc edit hco -n openshift-cnv kubevirt-hyperconverged
```

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  liveMigrationConfig: (1)
    bandwidthPerMigration: 64Mi
    completionTimeoutPerGiB: 800
    parallelMigrationsPerCluster: 5
    parallelOutboundMigrationsPerNode: 2
    progressTimeout: 150
```

1. Dans cet exemple, le tableau **spec.liveMigrationConfig** contient les valeurs par défaut de chaque champ.

NOTE

Vous pouvez rétablir la valeur par défaut de n'importe quel champ **spec.liveMigrationConfig** en supprimant cette paire clé/valeur et en enregistrant le fichier. Par exemple, supprimez **progressTimeout: <value>** pour rétablir la valeur par défaut **progressTimeout: 150**.

12.2.2. Limites et délais de migration en direct à l'échelle du cluster

Tableau 12.1. Paramètres de migration

Paramètres	Description	Défaut
parallelMigrationsPerCluster	Nombre de migrations exécutées en parallèle dans le cluster.	5
parallelOutboundMigrationsPerNode	Nombre maximal de migrations sortantes par nœud.	2
bandwidthPerMigration	Limite de bande passante de chaque migration, en MiB/s.	0 [1]
completionTimeoutPerGiB	La migration est annulée si elle n'est pas terminée dans ce délai, en secondes par gigaoctet de mémoire. Par exemple, une instance de machine virtuelle avec 6 Gigaoctets de mémoire est interrompue si la migration n'est pas terminée en 4800 secondes. Si l'adresse **Migration Method** est **BlockMigration**, la taille des disques de migration est incluse dans le calcul.	800
progressTimeout	La migration est annulée si la copie de la mémoire ne progresse pas dans ce délai, en secondes.	150

1. La valeur par défaut de **0** est illimitée.

12.3. MIGRATION D'UNE INSTANCE DE MACHINE VIRTUELLE VERS UN AUTRE NŒUD

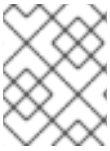
Lancer manuellement une migration en direct d'une instance de machine virtuelle vers un autre nœud à l'aide de la console Web ou de l'interface de ligne de commande.

NOTE

Si une machine virtuelle utilise un modèle de CPU hôte, vous pouvez effectuer une migration en direct de cette machine virtuelle uniquement entre les nœuds qui prennent en charge son modèle de CPU hôte.

12.3.1. Initier la migration en direct d'une instance de machine virtuelle dans la console web

Migrer une instance de machine virtuelle en cours d'exécution vers un autre nœud du cluster.



NOTE

L'action **Migrate** est visible par tous les utilisateurs, mais seuls les utilisateurs administrateurs peuvent initier une migration de machine virtuelle.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Vous pouvez lancer la migration à partir de cette page, ce qui facilite l'exécution d'actions sur plusieurs machines virtuelles sur la même page, ou à partir de la page **VirtualMachine details** où vous pouvez voir les détails complets de la machine virtuelle sélectionnée :
   - Cliquez sur le menu Options  à côté de la machine virtuelle et sélectionnez **Migrate**.
   - Cliquez sur le nom de la machine virtuelle pour ouvrir la page **VirtualMachine details** et cliquez sur **Actions → Migrate**.
3. Cliquez sur **Migrate** pour migrer la machine virtuelle vers un autre nœud.

12.3.2. Initier la migration en direct d'une instance de machine virtuelle dans le CLI

Initiez une migration en direct d'une instance de machine virtuelle en cours d'exécution en créant un objet **VirtualMachineInstanceMigration** dans le cluster et en référençant le nom de l'instance de machine virtuelle.

Procédure

1. Créez un fichier de configuration **VirtualMachineInstanceMigration** pour l'instance de machine virtuelle à migrer. Par exemple, **vmi-migrate.yaml**:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstanceMigration
metadata:
```

```
  name: migration-job
spec:
  vmiName: vmi-fedora
```

2. Créez l'objet dans le cluster en exécutant la commande suivante :

```
$ oc create -f vmi-migrate.yaml
```

L'objet **VirtualMachineInstanceMigration** déclenche une migration en direct de l'instance de machine virtuelle. Cet objet existe dans le cluster tant que l'instance de machine virtuelle est en cours d'exécution, sauf s'il est supprimé manuellement.

12.3.3. Ressources supplémentaires

- Suivi de la migration en direct
- Annulation de la migration en direct d'une instance de machine virtuelle

12.4. MIGRATION D'UNE MACHINE VIRTUELLE SUR UN RÉSEAU SUPPLÉMENTAIRE DÉDIÉ

Vous pouvez configurer un réseau Multus dédié pour la migration en direct. Un réseau dédié minimise les effets de la saturation du réseau sur les charges de travail des locataires pendant la migration en direct.

12.4.1. Configuration d'un réseau secondaire dédié pour la migration en direct des machines virtuelles

Pour configurer un réseau secondaire dédié pour la migration en direct, vous devez d'abord créer une définition d'attachement de réseau pont pour l'espace de noms **openshift-cnv** à l'aide de l'interface CLI. Ensuite, ajoutez le nom de l'objet **NetworkAttachmentDefinition** à la ressource personnalisée (CR) **HyperConverged**.

Conditions préalables

- You installed the OpenShift CLI (**oc**).
- Vous vous êtes connecté au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Le plugin Multus Container Network Interface (CNI) est installé sur le cluster.
- Chaque nœud de la grappe possède au moins deux cartes d'interface réseau (NIC), et les NIC à utiliser pour la migration en direct sont connectées au même VLAN.
- La machine virtuelle (VM) fonctionne avec la stratégie d'éviction **LiveMigrate**.

Procédure

1. Créer un manifeste **NetworkAttachmentDefinition**.

   Exemple de fichier de configuration

```
apiVersion: "k8s.cni.cncf.io/v1"
```

```
kind: NetworkAttachmentDefinition
metadata:
  name: my-secondary-network 1
  namespace: openshift-cnv 2
spec:
  config: '{
    "cniVersion": "0.3.1",
    "name": "migration-bridge",
    "type": "macvlan",
    "master": "eth1", 3
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts", 4
      "range": "10.200.5.0/24" 5
    }
  }'
```

1. Le nom de l'objet **NetworkAttachmentDefinition**.
2. L'espace de noms dans lequel réside l'objet **NetworkAttachmentDefinition**. Il doit s'agir de **openshift-cnv**.
3. Le nom du NIC à utiliser pour la migration en direct.
4. Le nom du plugin CNI qui fournit le réseau pour cette définition d'attachement réseau.
5. La plage d'adresses IP pour le réseau secondaire. Cette plage ne doit pas se chevaucher avec les adresses IP du réseau principal.

2. Ouvrez le CR **HyperConverged** dans votre éditeur par défaut en exécutant la commande suivante :

```
oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

3. Ajoutez le nom de l'objet **NetworkAttachmentDefinition** à la strophe **spec.liveMigrationConfig** de la CR **HyperConverged**. Par exemple :

Exemple de fichier de configuration

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  liveMigrationConfig:
    completionTimeoutPerGiB: 800
    network: my-secondary-network 1
    parallelMigrationsPerCluster: 5
    parallelOutboundMigrationsPerNode: 2
    progressTimeout: 150
...
```

1 Le nom de l'objet Multus **NetworkAttachmentDefinition** à utiliser pour les migrations en direct.

4. Enregistrez vos modifications et quittez l'éditeur. Les pods **virt-handler** redémarrent et se connectent au réseau secondaire.

Vérification

- Lorsque le nœud sur lequel tourne la machine virtuelle est placé en mode maintenance, la VM migre automatiquement vers un autre nœud du cluster. Vous pouvez vérifier que la migration s'est produite sur le réseau secondaire et non sur le réseau de pods par défaut en vérifiant l'adresse IP cible dans les métadonnées de l'instance de machine virtuelle (VMI).

```
oc get vmi <vmi_name> -o jsonpath='{.status.migrationState.targetNodeAddress}'
```

12.4.2. Sélection d'un réseau dédié à l'aide de la console web

Vous pouvez sélectionner un réseau dédié pour la migration en direct en utilisant la console web d'OpenShift Container Platform.

Conditions préalables

- Vous avez configuré un réseau Multus pour la migration en direct.

Procédure

1. Naviguez vers **Virtualization** > **Overview** dans la console web de OpenShift Container Platform.
2. Cliquez sur l'onglet **Settings** puis sur **Live migration**.
3. Sélectionnez le réseau dans la liste **Live migration network**.

12.4.3. Ressources supplémentaires

- Limites et délais de migration en direct

12.5. ANNULATION DE LA MIGRATION EN DIRECT D'UNE INSTANCE DE MACHINE VIRTUELLE

Annuler la migration en direct pour que l'instance de machine virtuelle reste sur le nœud d'origine.

Vous pouvez annuler une migration en direct à partir de la console web ou de l'interface CLI.

12.5.1. Annulation de la migration en direct d'une instance de machine virtuelle dans la console web

Vous pouvez annuler la migration en direct d'une instance de machine virtuelle dans la console web.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.

2. Cliquez sur le menu Options  à côté d'une machine virtuelle et sélectionnez **Cancel Migration**.

12.5.2. Annulation de la migration en direct d'une instance de machine virtuelle dans le CLI

Annuler la migration en direct d'une instance de machine virtuelle en supprimant l'objet **VirtualMachineInstanceMigration** associé à la migration.

Procédure

- Supprimez l'objet **VirtualMachineInstanceMigration** qui a déclenché la migration en direct, **migration-job** dans cet exemple :

```
$ oc delete vmim migration-job
```

12.6. CONFIGURATION DE LA STRATÉGIE D'ÉVICTION DES MACHINES VIRTUELLES

La stratégie d'éviction **LiveMigrate** garantit qu'une instance de machine virtuelle n'est pas interrompue si le nœud est placé en maintenance ou vidangé. Les instances de machines virtuelles avec cette stratégie d'éviction seront migrées en direct vers un autre nœud.

12.6.1. Configurer des machines virtuelles personnalisées avec la stratégie d'éviction LiveMigration

Vous ne devez configurer la stratégie d'éviction **LiveMigration** que sur les machines virtuelles personnalisées. Les modèles communs ont cette stratégie d'éviction configurée par défaut.

Procédure

1. Ajoutez l'option **evictionStrategy: LiveMigrate** à la section **spec.template.spec** du fichier de configuration de la machine virtuelle. Cet exemple utilise **oc edit** pour mettre à jour l'extrait correspondant du fichier de configuration **VirtualMachine**:

```
$ oc edit vm <custom-vm> -n <my-namespace>
```

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: custom-vm
spec:
  template:
    spec:
      evictionStrategy: LiveMigrate
...
```

2. Redémarrez la machine virtuelle pour que la mise à jour prenne effet :

```
virtctl restart <custom-vm> -n <my-namespace>
```

12.7. CONFIGURATION DES POLITIQUES DE MIGRATION EN DIRECT

Vous pouvez définir différentes configurations de migration pour des groupes spécifiques d'instances de machines virtuelles (VMI) à l'aide d'une stratégie de migration en direct.

IMPORTANT

La politique de migration en direct est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

Pour configurer une politique de migration en direct à l'aide de la console web, voir la documentation de la page MigrationPolicies.

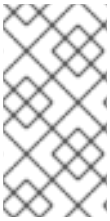
12.7.1. Configurer une politique de migration en direct à partir de la ligne de commande

Utilisez la définition de ressource personnalisée (CRD) **MigrationPolicy** pour définir des politiques de migration pour un ou plusieurs groupes d'instances de machines virtuelles (VMI) sélectionnées.

Vous pouvez spécifier des groupes de VMI en utilisant une combinaison des éléments suivants :

- Étiquettes d'instances de machines virtuelles telles que **size**, **os**, **gpu**, et autres étiquettes VMI.
- Étiquettes d'espace de noms telles que **priority**, **bandwidth**, **hpc-workload**, et autres étiquettes d'espace de noms.

Pour que la politique s'applique à un groupe spécifique de VMI, toutes les étiquettes du groupe de VMI doivent correspondre aux étiquettes de la politique.



NOTE

Si plusieurs politiques de migration en direct s'appliquent à une IMV, la politique ayant le plus grand nombre d'étiquettes correspondantes est prioritaire. Si plusieurs politiques répondent à ce critère, elles sont triées par ordre lexicographique des clés d'étiquettes correspondantes, et la première dans cet ordre est prioritaire.

Procédure

1. Créez un CRD **MigrationPolicy** pour le groupe de VMI que vous avez spécifié. L'exemple YAML suivant configure un groupe avec les étiquettes **hpc-workloads:true**, **xyz-workloads-type: ""**, **workload-type: db**, et **operating-system: ""**:

```
apiVersion: migrations.kubevirt.io/v1alpha1
kind: MigrationPolicy
metadata:
```

```
  name: my-awesome-policy
spec:
  # Migration Configuration
  allowAutoConverge: true
  bandwidthPerMigration: 217Ki
  completionTimeoutPerGiB: 23
  allowPostCopy: false

  # Matching to VMIs
  selectors:
    namespaceSelector: (1)
      hpc-workloads: "True"
      xyz-workloads-type: ""
    virtualMachineInstanceSelector: (2)
      workload-type: "db"
      operating-system: ""
```

(1) Utilisez **namespaceSelector** pour définir un groupe de VMI à l'aide d'étiquettes d'espace de noms.

(2) Utilisez **virtualMachineInstanceSelector** pour définir un groupe de VMI à l'aide d'étiquettes VMI.

CHAPITRE 13. MAINTENANCE DES NŒUDS

13.1. À PROPOS DE LA MAINTENANCE DES NŒUDS

13.1.1. A propos du mode de maintenance des nœuds

Les nœuds peuvent être placés en mode maintenance à l'aide de l'utilitaire **oc adm** ou des ressources personnalisées (CR) de **NodeMaintenance**.

NOTE

Le **node-maintenance-operator** (NMO) n'est plus livré avec OpenShift Virtualization. Il est désormais disponible pour être déployé en tant qu'opérateur autonome depuis le site **OperatorHub** dans la console web d'OpenShift Container Platform, ou en utilisant la CLI d'OpenShift (**oc**).

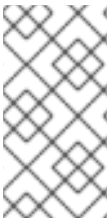
Le placement d'un nœud en maintenance indique que le nœud est inutilisable et draine toutes les machines virtuelles et tous les pods qui s'y trouvent. Les instances de machines virtuelles qui ont une stratégie d'éviction **LiveMigrate** sont migrées en direct vers un autre nœud sans perte de service. Cette stratégie d'éviction est configurée par défaut dans les machines virtuelles créées à partir de modèles communs, mais doit être configurée manuellement pour les machines virtuelles personnalisées.

Les instances de machines virtuelles sans stratégie d'éviction sont arrêtées. Les machines virtuelles dont l'adresse **RunStrategy** est **Running** ou **RerunOnFailure** sont recréées sur un autre nœud. Les machines virtuelles dont l'adresse **RunStrategy** est **Manual** ne sont pas redémarrées automatiquement.

IMPORTANT

Les machines virtuelles doivent disposer d'une revendication de volume persistant (PVC) avec un mode d'accès partagé **ReadWriteMany** (RWX) pour être migrées en direct.

L'opérateur de maintenance des nœuds surveille les CR **NodeMaintenance** nouveaux ou supprimés. Lorsqu'un nouveau CR **NodeMaintenance** est détecté, aucune nouvelle charge de travail n'est programmée et le nœud est isolé du reste du cluster. Tous les pods qui peuvent être expulsés le sont du nœud. Lorsqu'un CR **NodeMaintenance** est supprimé, le nœud référencé dans le CR est rendu disponible pour de nouvelles charges de travail.



NOTE

L'utilisation d'un CR **NodeMaintenance** pour les tâches de maintenance des nœuds permet d'obtenir les mêmes résultats que les commandes **oc adm cordon** et **oc adm drain** à l'aide du traitement standard des ressources personnalisées d'OpenShift Container Platform.

13.1.2. Maintenance des nœuds métalliques nus

Lorsque vous déployez OpenShift Container Platform sur une infrastructure bare metal, il y a des considérations supplémentaires qui doivent être prises en compte par rapport au déploiement sur une infrastructure cloud. Contrairement aux environnements cloud où les nœuds de cluster sont considérés comme éphémères, le réapprovisionnement d'un nœud bare metal nécessite beaucoup plus de temps et d'efforts pour les tâches de maintenance.

Lorsqu'un nœud bare metal tombe en panne, par exemple en cas d'erreur fatale du noyau ou de défaillance matérielle d'une carte NIC, les charges de travail sur le nœud en panne doivent être redémarrées ailleurs dans la grappe pendant que le nœud défectueux est réparé ou remplacé. Le mode de maintenance des nœuds permet aux administrateurs de clusters de mettre les nœuds hors tension de manière élégante, en déplaçant les charges de travail vers d'autres parties du cluster et en veillant à ce que les charges de travail ne soient pas interrompues. Des informations détaillées sur la progression et l'état des nœuds sont fournies pendant la maintenance.

13.1.3. Ressources supplémentaires

- Installation de l'opérateur de maintenance de nœuds à l'aide de la CLI
- Mise en mode maintenance d'un nœud
- Reprise d'un nœud en mode maintenance
- À propos des stratégies d'exécution pour les machines virtuelles
- Migration en direct de la machine virtuelle
- Configuration de la stratégie d'éviction des machines virtuelles

13.2. RENOUVELLEMENT AUTOMATIQUE DES CERTIFICATS TLS

Tous les certificats TLS pour les composants d'OpenShift Virtualization sont renouvelés et font l'objet d'une rotation automatique. Il n'est pas nécessaire de les rafraîchir manuellement.

13.2.1. Calendrier de renouvellement automatique des certificats TLS

Les certificats TLS sont automatiquement supprimés et remplacés selon le calendrier suivant :

- Les certificats KubeVirt sont renouvelés quotidiennement.
- Les certificats de contrôleur d'importation de données conteneurisées (CDI) sont renouvelés tous les 15 jours.
- Les certificats de pool MAC sont renouvelés chaque année.

La rotation automatique des certificats TLS ne perturbe aucune opération. Par exemple, les opérations suivantes continuent de fonctionner sans interruption :

- Migrations
- Chargement d'images
- Connexions VNC et consoles

13.3. GESTION DE L'ÉTIQUETAGE DES NŒUDS POUR LES MODÈLES DE CPU OBSOLÈTES

Vous pouvez planifier une machine virtuelle (VM) sur un nœud à condition que le modèle de CPU de la VM et la stratégie soient pris en charge par le nœud.

13.3.1. À propos de l'étiquetage des nœuds pour les modèles de CPU obsolètes

L'opérateur de virtualisation OpenShift utilise une liste prédéfinie de modèles de CPU obsolètes pour s'assurer qu'un nœud ne prend en charge que des modèles de CPU valides pour les VM planifiées.

Par défaut, les modèles de CPU suivants sont éliminés de la liste des étiquettes générées pour le nœud :

Exemple 13.1. Modèles de CPU obsolètes

```
"486"
Conroe
athlon
core2duo
coreduo
kvm32
kvm64
n270
pentium
pentium2
pentium3
pentiumpro
phenom
qemu32
qemu64
```

Cette liste prédéfinie n'est pas visible dans le CR **HyperConverged**. Vous ne pouvez pas *remove* les modèles de CPU de cette liste, mais vous pouvez ajouter à la liste en modifiant le champ **spec.obsoleteCPUs.cpuModels** du CR **HyperConverged**.

13.3.2. À propos de l'étiquetage des nœuds pour les caractéristiques de l'unité centrale

Grâce au processus d'itération, les caractéristiques de l'unité centrale de base dans le modèle d'unité centrale minimale sont éliminées de la liste des étiquettes générées pour le nœud.

Par exemple :

- Un environnement peut avoir deux modèles de CPU pris en charge : **Penryn** et **Haswell**.
- Si **Penryn** est spécifié comme modèle d'unité centrale pour **minCPU**, chaque caractéristique de base de l'unité centrale pour **Penryn** est comparée à la liste des caractéristiques de l'unité centrale prises en charge par **Haswell**.

  Exemple 13.2. Fonctionnalités de l'unité centrale prises en charge par Penryn

  ```
  apic
  clflush
  cmov
  cx16
  cx8
  de
  fpu
  fxsr
  lahf_lm
  lm
  mca
  ```

```
mce
mmx
msr
mtrr
nx
pae
pat
pge
pni
pse
pse36
sep
sse
sse2
sse4.1
ssse3
syscall
tsc
```

Exemple 13.3. Fonctionnalités de l'unité centrale prises en charge par Haswell

```
aes
apic
avx
avx2
bmi1
bmi2
clflush
cmov
cx16
cx8
de
erms
fma
fpu
fsgsbase
fxsr
hle
invpcid
lahf_lm
lm
mca
mce
mmx
movbe
msr
mtrr
nx
pae
pat
pcid
pclmuldq
pge
pni
```

```
popcnt
pse
pse36
rdtscp
rtm
sep
smep
sse
sse2
sse4.1
sse4.2
ssse3
syscall
tsc
tsc-deadline
x2apic
xsave
```

- Si les sites **Penryn** et **Haswell** prennent tous deux en charge une fonctionnalité spécifique de l'unité centrale, aucune étiquette n'est créée pour cette fonctionnalité. Des étiquettes sont générées pour les fonctionnalités de l'unité centrale qui sont prises en charge uniquement par **Haswell** et non par **Penryn**.

Exemple 13.4. Étiquettes de nœuds créées pour les caractéristiques de l'unité centrale après itération

```
aes
avx
avx2
bmi1
bmi2
erms
fma
fsgsbase
hle
invpcid
movbe
pcid
pclmuldq
popcnt
rdtscp
rtm
sse4.2
tsc-deadline
x2apic
xsave
```

13.3.3. Configuration des modèles de CPU obsolètes

Vous pouvez configurer une liste de modèles de CPU obsolètes en modifiant la ressource personnalisée (CR) **HyperConverged**.

Procédure

- Modifiez la ressource personnalisée **HyperConverged** en spécifiant les modèles de CPU obsolètes dans le tableau **obsoleteCPUs**. Par exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  obsoleteCPUs:
    cpuModels: 1
      - "<obsolete_cpu_1>"
      - "<obsolete_cpu_2>"
    minCPUModel: "<minimum_cpu_model>" 2
```

1. Remplacer les valeurs de l'exemple dans le tableau **cpuModels** par des modèles de CPU obsolètes. Toute valeur spécifiée est ajoutée à une liste prédéfinie de modèles d'unités centrales obsolètes. La liste prédéfinie n'est pas visible dans le CR.

2. Remplacez cette valeur par le modèle de CPU minimum que vous souhaitez utiliser pour les fonctionnalités de base du CPU. Si vous ne spécifiez pas de valeur, **Penryn** est utilisé par défaut.

13.4. EMPÊCHER LE RAPPROCHEMENT DES NŒUDS

Utilisez l'annotation **skip-node** pour empêcher **node-labeller** de réconcilier un nœud.

13.4.1. Utilisation de l'annotation skip-node

Si vous souhaitez que **node-labeller** saute un nœud, annotez ce nœud à l'aide de l'interface CLI de **oc**.

Conditions préalables

- Vous avez installé l'OpenShift CLI (**oc**).

Procédure

- Annotez le nœud que vous souhaitez ignorer en exécutant la commande suivante :

```
$ oc annotate node <node_name> node-labeller.kubevirt.io/skip-node=true 1
```

1. Remplacez **<node_name>** par le nom du nœud à ignorer.

La réconciliation reprend au cycle suivant après que l'annotation du nœud a été supprimée ou mise à faux.

13.4.2. Ressources supplémentaires

- Gestion de l'étiquetage des nœuds pour les modèles de CPU obsolètes

CHAPITRE 14. JOURNALISATION, ÉVÉNEMENTS ET SURVEILLANCE

14.1. VUE D'ENSEMBLE DE LA VIRTUALISATION

La page **Virtualization Overview** offre une vue d'ensemble des ressources de virtualisation, des détails, de l'état et des principaux consommateurs :

- L'onglet **Overview** affiche les ressources **Getting started**, les détails, l'inventaire, les alertes et d'autres informations sur votre environnement OpenShift Virtualization.
- L'onglet **Top consumers** affiche l'utilisation élevée d'une ressource spécifique par les projets, les machines virtuelles ou les nœuds.
- L'onglet **Migrations** affiche l'état des migrations en cours.
- L'onglet **Settings** affiche les paramètres de l'ensemble du cluster, y compris les paramètres de migration en direct et les autorisations des utilisateurs.

En obtenant un aperçu de la santé globale d'OpenShift Virtualization, vous pouvez déterminer si une intervention est nécessaire pour résoudre des problèmes spécifiques identifiés en examinant les données.

14.1.1. Examen des principaux consommateurs

Vous pouvez afficher les principaux consommateurs de ressources pour un projet, une machine virtuelle ou un nœud sélectionné dans l'onglet **Top consumers** de la page **Virtualization Overview**.

Conditions préalables

- Vous devez avoir accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Pour utiliser la métrique **vCPU wait** dans l'onglet **Top consumers**, vous devez appliquer l'argument du noyau **schedstats=enable** à l'objet **MachineConfig**.

Procédure

1. Dans la perspective **Administrator** de la console web OpenShift Container Platform, naviguez vers **Virtualization → Overview**.
2. Cliquez sur l'onglet **Top consumers**.
3. Facultatif : vous pouvez filtrer les résultats en sélectionnant une période ou en sélectionnant les 5 ou 10 premiers consommateurs.

14.1.2. Ressources supplémentaires

- Ajout d'arguments de noyau aux nœuds
- Aperçu de la surveillance
- Examen des tableaux de bord de suivi
- Tableaux de bord

14.2. VISUALISATION DES JOURNAUX D'OPENSHIFT VIRTUALIZATION

Vous pouvez consulter les logs des composants d'OpenShift Virtualization et des machines virtuelles en utilisant la console web ou le CLI **oc**. Vous pouvez récupérer les journaux des machines virtuelles à partir du pod **virt-launcher**. Pour contrôler la verbosité des journaux, modifiez la ressource personnalisée **HyperConverged**.

14.2.1. Visualiser les logs d'OpenShift Virtualization avec le CLI

Configurez la verbosité des logs pour les composants d'OpenShift Virtualization en éditant la ressource personnalisée (CR) **HyperConverged**. Ensuite, affichez les journaux pour les pods de composants en utilisant l'outil CLI **oc**.

Procédure

1. Pour définir la verbosité des journaux pour des composants spécifiques, ouvrez le CR **HyperConverged** dans votre éditeur de texte par défaut en exécutant la commande suivante :

```
$ oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

2. Définissez le niveau de journalisation pour un ou plusieurs composants en modifiant la strophe **spec.logVerbosityConfig**. Par exemple :

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
spec:
  logVerbosityConfig:
    kubevirt:
      virtAPI: 5 (1)
      virtController: 4
      virtHandler: 3
      virtLauncher: 2
      virtOperator: 6
```

(1) La valeur de la verbosité du journal doit être un nombre entier compris dans l'intervalle **1–9**, où un nombre plus élevé indique un journal plus détaillé. Dans cet exemple, les journaux du composant **virtAPI** sont exposés si leur niveau de priorité est égal ou supérieur à **5**.

3. Appliquez vos modifications en enregistrant et en quittant l'éditeur.

4. Affichez une liste de pods dans l'espace de noms OpenShift Virtualization en exécutant la commande suivante :

```
$ oc get pods -n openshift-cnv
```

Exemple 14.1. Exemple de sortie

```
NAME                               READY   STATUS    RESTARTS   AGE
disks-images-provider-7gqbc        1/1     Running   0          32m
disks-images-provider-vg4kx        1/1     Running   0          32m
virt-api-57fcc4497b-7qfmc          1/1     Running   0          31m
```

```
virt-api-57fcc4497b-tx9nc          1/1     Running   0          31m
virt-controller-76c784655f-7fp6m   1/1     Running   0          30m
virt-controller-76c784655f-f4pbd   1/1     Running   0          30m
virt-handler-2m86x                 1/1     Running   0          30m
virt-handler-9qs6z                 1/1     Running   0          30m
virt-operator-7ccfdbf65f-q5snk     1/1     Running   0          32m
virt-operator-7ccfdbf65f-vllz8     1/1     Running   0          32m
```

5. Pour afficher les journaux d'un pod de composants, exécutez la commande suivante :

```
$ oc logs -n openshift-cnv <nom_du_pod>
```

Par exemple :

```
$ oc logs -n openshift-cnv virt-handler-2m86x
```

NOTE

Si un pod ne démarre pas, vous pouvez utiliser l'option **--previous** pour afficher les journaux de la dernière tentative.

Pour surveiller la sortie des journaux en temps réel, utilisez l'option **-f**.

Exemple 14.2. Exemple de sortie

```
{"component":"virt-handler","level":"info","msg":"set verbosity to 2","pos":"virt-
handler.go:453","timestamp":"2022-04-17T08:58:37.373695Z"}
{"component":"virt-handler","level":"info","msg":"set verbosity to 2","pos":"virt-
handler.go:453","timestamp":"2022-04-17T08:58:37.373726Z"}
{"component":"virt-handler","level":"info","msg":"setting rate limiter to 5 QPS and 10
Burst","pos":"virt-handler.go:462","timestamp":"2022-04-17T08:58:37.373782Z"}
{"component":"virt-handler","level":"info","msg":"CPU features of a minimum baseline CPU
model: map[apic:true clflush:true cmov:true cx16:true cx8:true de:true fpu:true fxsr:true
lahf_lm:true lm:true mca:true mce:true mmx:true msr:true mtrr:true nx:true pae:true
pat:true pge:true pni:true pse:true pse36:true sep:true sse:true sse2:true sse4.1:true
ssse3:true syscall:true tsc:true]","pos":"cpu_plugin.go:96","timestamp":"2022-04-
17T08:58:37.390221Z"}
{"component":"virt-handler","level":"warning","msg":"host model mode is expected to
contain only one model","pos":"cpu_plugin.go:103","timestamp":"2022-04-
17T08:58:37.390263Z"}
{"component":"virt-handler","level":"info","msg":"node-labeller is
running","pos":"node_labeller.go:94","timestamp":"2022-04-17T08:58:37.391011Z"}
```

14.2.2. Visualisation des journaux des machines virtuelles dans la console web

Obtenir les journaux de la machine virtuelle à partir du module de lancement de la machine virtuelle associée.

Procédure

1. Dans la console OpenShift Container Platform, cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Details**.
4. Cliquez sur le pod **virt-launcher-<name>** dans la section **Pod** pour ouvrir la page **Pod details**.
5. Cliquez sur l'onglet **Logs** pour afficher les journaux de pods.

14.2.3. Messages d'erreur courants

Les messages d'erreur suivants peuvent apparaître dans les journaux d'OpenShift Virtualization :

ErrImagePull ou **ImagePullBackOff**

Indique une configuration de déploiement incorrecte ou des problèmes avec les images référencées.

14.3. VISUALISATION DES ÉVÉNEMENTS

14.3.1. À propos des événements liés aux machines virtuelles

Les événements d'OpenShift Container Platform sont des enregistrements d'informations importantes sur le cycle de vie d'un espace de noms et sont utiles pour surveiller et résoudre les problèmes de planification, de création et de suppression des ressources.

OpenShift Virtualization ajoute des événements pour les machines virtuelles et les instances de machines virtuelles. Ceux-ci peuvent être consultés à partir de la console web ou de la CLI.

Voir aussi : Affichage des informations sur les événements système dans un cluster OpenShift Container Platform.

14.3.2. Visualisation des événements d'une machine virtuelle dans la console web

Vous pouvez visualiser les événements de streaming pour une machine virtuelle en cours d'exécution sur la page **VirtualMachine details** de la console Web.

Procédure

1. Cliquez sur **Virtualization → VirtualMachines** dans le menu latéral.
2. Sélectionnez une machine virtuelle pour ouvrir la page **VirtualMachine details**.
3. Cliquez sur l'onglet **Events** pour afficher les événements de flux pour la machine virtuelle.
   - Le bouton ▮▮ met en pause le flux d'événements.
   - Le bouton ▶ permet de reprendre un flux d'événements en pause.

14.3.3. Visualisation des événements de l'espace de noms dans l'interface de programmation

Utilisez le client OpenShift Container Platform pour obtenir les événements d'un espace de noms.

Procédure

- Dans l'espace de noms, utilisez la commande **oc get**:

```
$ oc get events
```

14.3.4. Visualisation des événements de ressources dans l'interface de ligne de commande

Les événements sont inclus dans la description de la ressource, que vous pouvez obtenir en utilisant le client OpenShift Container Platform.

Procédure

- Dans l'espace de noms, utilisez la commande **oc describe**. L'exemple suivant montre comment obtenir les événements pour une machine virtuelle, une instance de machine virtuelle et le pod virt-launcher pour une machine virtuelle :

```
oc describe vm <vm> $ oc describe vm <vm>
```

```
$ oc describe vmi <vmi>
```

```
$ oc describe pod virt-launcher-<name>
```

14.4. SUIVI DE LA MIGRATION EN DIRECT

Vous pouvez surveiller la progression de la migration en direct à partir de la console web ou de l'interface de commande en ligne.

14.4.1. Suivi de la migration en direct à l'aide de la console web

Vous pouvez suivre la progression de toutes les migrations en direct dans l'ongletOverview → Migrations de la console web.

Vous pouvez afficher les mesures de migration d'une machine virtuelle sur l'ongletVirtualMachine details → Metrics dans la console Web.

14.4.2. Surveillance de la migration en direct d'une instance de machine virtuelle dans le CLI

L'état de la migration de la machine virtuelle est stocké dans le composant **Status** de la configuration **VirtualMachineInstance**.

Procédure

- Utilisez la commande **oc describe** sur l'instance de machine virtuelle qui migre :

```
$ oc describe vmi vmi-fedora
```

Exemple de sortie

```
...
Status:
  Conditions:
    Last Probe Time:       <nil>
    Last Transition Time:  <nil>
    Status:                True
    Type:                  LiveMigratable
  Migration Method:  LiveMigration
  Migration State:
    Completed:                true
    End Timestamp:            2018-12-24T06:19:42Z
    Migration UID:            d78c8962-0743-11e9-a540-fa163e0c69f1
    Source Node:              node2.example.com
    Start Timestamp:          2018-12-24T06:19:35Z
    Target Node:              node1.example.com
    Target Node Address:      10.9.0.18:43891
    Target Node Domain Detected:  true
```

14.4.3. Metrics

Vous pouvez utiliser les requêtes Prometheus pour surveiller la migration en direct.

14.4.3.1. Mesures de migration en temps réel

Les paramètres suivants peuvent être interrogés pour connaître l'état de la migration en temps réel :

kubevirt_migrate_vmi_data_processed_bytes

La quantité de données du système d'exploitation invité (OS) qui a migré vers la nouvelle machine virtuelle (VM). Type : Jauge.

kubevirt_migrate_vmi_data_remaining_bytes

Quantité de données du système d'exploitation invité restant à migrer. Type : Jauge.

kubevirt_migrate_vmi_dirty_memory_rate_bytes

La vitesse à laquelle la mémoire devient sale dans le système d'exploitation invité. La mémoire sale est constituée de données qui ont été modifiées mais qui n'ont pas encore été écrites sur le disque. Type : Jauge.

kubevirt_migrate_vmi_pending_count

Le nombre de migrations en attente. Type : Jauge.

kubevirt_migrate_vmi_scheduling_count

Le nombre de migrations d'ordonnancement. Type : Jauge.

kubevirt_migrate_vmi_running_count

Le nombre de migrations en cours. Type : Jauge.

kubevirt_migrate_vmi_succeeded

Le nombre de migrations effectuées avec succès. Type : Jauge.

kubevirt_migrate_vmi_failed

Le nombre de migrations qui ont échoué. Type : Jauge.

14.5. DIAGNOSTIC DES VOLUMES DE DONNÉES À L'AIDE D'ÉVÉNEMENTS ET DE CONDITIONS

La commande **oc describe** permet d'analyser et de résoudre les problèmes liés aux volumes de données.

14.5.1. Conditions et événements

Diagnostiquez les problèmes de volume de données en examinant la sortie des sections **Conditions** et **Events** générées par la commande :

```
oc describe dv <DataVolume> $ oc describe dv <DataVolume>
```

Types La section **Conditions** contient trois sites Internet qui s'affichent :

- **Bound**
- **Running**
- **Ready**

La section **Events** fournit les informations supplémentaires suivantes :

- **Type** de l'événement
- **Reason** pour l'enregistrement
- **Source** de l'événement
- **Message** contenant des informations de diagnostic supplémentaires.

La sortie de **oc describe** ne contient pas toujours **Events**.

Un événement est généré lorsque **Status**, **Reason** ou **Message** change. Les conditions et les événements réagissent aux changements d'état du volume de données.

Par exemple, si l'URL est mal orthographié lors d'une opération d'importation, l'importation génère un message 404. Ce changement de message génère un événement avec un motif. La sortie de la section **Conditions** est également mise à jour.

14.5.2. Analyse des volumes de données à l'aide de conditions et d'événements

En inspectant les sections **Conditions** et **Events** générées par la commande **describe**, vous déterminez l'état du volume de données par rapport aux réclamations de volume persistantes (PVC), et si une opération est en cours ou terminée. Vous pouvez également recevoir des messages contenant des détails spécifiques sur l'état du volume de données et sur la manière dont il est arrivé à son état actuel.

Il existe de nombreuses combinaisons différentes de conditions. Chacune doit être évaluée dans son contexte particulier.

Des exemples de différentes combinaisons sont présentés ci-dessous.

- **Bound** - Un PVC lié avec succès s'affiche dans cet exemple.
  Notez que le **Type** est **Bound**, donc le **Status** est **True**. Si le PVC n'est pas lié, le **Status** est **False**.

  Lorsque le PVC est lié, un événement est généré indiquant que le PVC est lié. Dans ce cas, **Reason** est **Bound** et **Status** est **True**. Le **Message** indique à quel PVC appartient le volume de données.

Messagela section **Events** fournit des informations complémentaires, notamment sur la durée de la liaison (**Age**) et sur la ressource (**From**), en l'occurrence **datavolume-controller**:

Exemple de sortie

```
Status:
 Conditions:
  Last Heart Beat Time:  2020-07-15T03:58:24Z
  Last Transition Time:  2020-07-15T03:58:24Z
  Message:               PVC win10-rootdisk Bound
  Reason:                Bound
  Status:                True
  Type:                  Bound

 Events:
  Type    Reason    Age    From                Message
  ----    ------    ----   ----                -------
  Normal  Bound     24s    datavolume-controller  PVC example-dv Bound
```

- **Running** - Dans ce cas, notez que **Type** est **Running** et **Status** est **False**, ce qui indique qu'un événement s'est produit et a entraîné l'échec d'une tentative d'opération, faisant passer l'état de **True** à **False**.
  Notez toutefois que **Reason** correspond à **Completed** et que le champ **Message** correspond à **Import Complete**.

  Dans la section **Events**, les sections **Reason** et **Message** contiennent des informations de dépannage supplémentaires sur l'échec de l'opération. Dans cet exemple, le site **Message** affiche une impossibilité de se connecter en raison d'un problème **404**, répertorié dans le premier **Warning** de la section **Events**.

  Ces informations vous permettent de conclure qu'une opération d'importation était en cours d'exécution, ce qui créait de la contention pour les autres opérations qui tentaient d'accéder au volume de données :

  Exemple de sortie

```
Status:
  Conditions:
   Last Heart Beat Time:  2020-07-15T04:31:39Z
   Last Transition Time:  2020-07-15T04:31:39Z
   Message:               Import Complete
   Reason:                Completed
   Status:                False
   Type:                  Running

  Events:
   Type     Reason          Age              From                Message
   ----     ------          ----             ----                -------
   Warning  Error           12s (x2 over 14s)  datavolume-controller  Unable to connect
   to http data source: expected status code 200, got 404. Status: 404 Not Found
```

- **Ready** - Si **Type** est **Ready** et **Status** est **True**, le volume de données est prêt à être utilisé, comme dans l'exemple suivant. Si le volume de données n'est pas prêt à être utilisé, l'adresse **Status** est **False**:

  Exemple de sortie

Exemple de sortie

```
Status:
  Conditions:
    Last Heart Beat Time: 2020-07-15T04:31:39Z
    Last Transition Time:  2020-07-15T04:31:39Z
    Status:               True
    Type:                 Ready
```

14.6. AFFICHAGE D'INFORMATIONS SUR LES CHARGES DE TRAVAIL DES MACHINES VIRTUELLES

Vous pouvez afficher des informations de haut niveau sur vos machines virtuelles en utilisant le tableau de bord **Virtual Machines** dans la console web d'OpenShift Container Platform.

14.6.1. Le tableau de bord Machines virtuelles

Accédez aux machines virtuelles (VM) depuis la console web d'OpenShift Container Platform en naviguant vers la page **Virtualization → VirtualMachines** et en cliquant sur une machine virtuelle (VM) pour afficher la page **VirtualMachine details**.

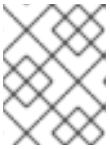
L'onglet **Overview** affiche les cartes suivantes :

- **Details** fournit des informations d'identification sur la machine virtuelle, notamment
  - Nom
  - Statut
  - Date de création
  - Système d'exploitation
  - Unité centrale et mémoire
  - Nom d'hôte
  - Modèle

  Si la VM est en cours d'exécution, une fenêtre de prévisualisation VNC est active et un lien permet d'ouvrir la console web VNC. Le menu **Options** sur la carte **Details** permet d'arrêter ou de mettre en pause la VM et de copier la commande **ssh over nodeport** pour le tunnel SSH.
- **Alerts** répertorie les alertes VM selon trois niveaux de gravité :
  - Critique
  - Avertissement
  - Info
- **Snapshots** fournit des informations sur les instantanés de VM et sur la possibilité de prendre un instantané. Pour chaque instantané répertorié, la carte **Snapshots** comprend :

- Indicateur visuel de l'état de l'instantané, indiquant s'il a été créé avec succès, s'il est toujours en cours ou s'il a échoué.
- Un menu **Options** avec des options pour restaurer ou supprimer l'instantané

- **Network interfaces** fournit des informations sur les interfaces réseau de la VM, notamment :
  - Nom (réseau et type)
  - L'adresse IP, avec la possibilité de copier l'adresse IP dans le presse-papiers
- **Disks** liste les détails des disques VM, y compris :
  - Nom
  - Conduire
  - Taille
- **Utilization** comprend des graphiques qui affichent des données d'utilisation pour :
  - UNITÉ CENTRALE
  - Mémoire
  - Stockage
  - Transfert de réseau



NOTE

Utilisez la liste déroulante pour choisir une durée pour les données d'utilisation. Les options disponibles sont **5 minutes**, **1 hour**, **6 hours**, et **24 hours**.

- **Hardware Devices** fournit des informations sur le GPU et les périphériques hôtes, notamment :
  - Nom de la ressource
  - Nom de l'appareil matériel

14.7. SURVEILLANCE DE LA SANTÉ DES MACHINES VIRTUELLES

Une instance de machine virtuelle (IMV) peut devenir malsaine en raison de problèmes transitoires tels qu'une perte de connectivité, des blocages ou des problèmes liés à des dépendances externes. Un contrôle de santé effectue périodiquement des diagnostics sur une IMV en utilisant n'importe quelle combinaison de sondes de disponibilité et d'état d'avancement.

14.7.1. À propos des sondes de disponibilité et d'intégrité

Utilisez des sondes de disponibilité et de vivacité pour détecter et gérer les instances de machines virtuelles (VMI) en mauvaise santé. Vous pouvez inclure une ou plusieurs sondes dans la spécification de la VMI pour vous assurer que le trafic n'atteint pas une VMI qui n'est pas prête à l'accueillir et qu'une nouvelle instance est créée lorsqu'une VMI ne répond plus.

Une adresse *readiness probe* détermine si une IMV est prête à accepter des demandes de service. Si la sonde échoue, la VMI est retirée de la liste des points d'extrémité disponibles jusqu'à ce que la VMI soit prête.

Un site *liveness probe* détermine si une IMV est réactive. Si la sonde échoue, la VMI est supprimée et une nouvelle instance est créée pour rétablir la réactivité.

Vous pouvez configurer les tests de disponibilité et d'accessibilité en définissant les champs **spec.readinessProbe** et **spec.livenessProbe** de l'objet **VirtualMachineInstance**. Ces champs prennent en charge les tests suivants :

HTTP GET

La sonde détermine l'état de santé de l'IMV à l'aide d'un crochet Web. Le test est réussi si le code de réponse HTTP est compris entre 200 et 399. Vous pouvez utiliser un test HTTP GET avec des applications qui renvoient des codes d'état HTTP lorsqu'elles sont complètement initialisées.

Socket TCP

La sonde tente d'ouvrir une connexion avec l'IMV. L'IMV n'est considérée comme saine que si la sonde peut établir une connexion. Vous pouvez utiliser un test de socket TCP avec des applications qui ne commencent à écouter qu'une fois l'initialisation terminée.

Agent invité ping

La sonde utilise la commande **guest-ping** pour déterminer si l'agent invité QEMU est exécuté sur la machine virtuelle.

14.7.2. Définition d'une sonde de préparation HTTP

Définissez une sonde de préparation HTTP en définissant le champ **spec.readinessProbe.httpGet** de la configuration de l'instance de machine virtuelle (VMI).

Procédure

1. Inclure les détails de la sonde de disponibilité dans le fichier de configuration de l'IMV.

Exemple de sonde de préparation avec un test HTTP GET

```
# ...
spec:
  readinessProbe:
    httpGet: 1
      port: 1500 2
      path: /healthz 3
      httpHeaders:
      - name: Custom-Header
        value: Awesome
    initialDelaySeconds: 120 4
    periodSeconds: 20 5
    timeoutSeconds: 10 6
    failureThreshold: 3 7
    successThreshold: 3 8
# ...
```

1 La requête HTTP GET à effectuer pour se connecter à l'IMV.

2 Le port de l'IMV que la sonde interroge. Dans l'exemple ci-dessus, la sonde interroge le port 1500.

3 Le chemin d'accès au serveur HTTP. Dans l'exemple ci-dessus, si le gestionnaire du chemin /healthz du serveur renvoie un code de réussite, l'IMV est considérée comme saine. Si le gestionnaire renvoie un code d'échec, l'IMV est supprimée de la liste des points d'extrémité disponibles.

4 Temps, en secondes, écoulé entre le démarrage de l'IMV et le lancement de la sonde d'état de préparation.

5 Délai, en secondes, entre l'exécution des sondes. Le délai par défaut est de 10 secondes. Cette valeur doit être supérieure à **timeoutSeconds**.

6 Le nombre de secondes d'inactivité après lesquelles la sonde s'arrête et la VMI est considérée comme ayant échoué. La valeur par défaut est 1. Cette valeur doit être inférieure à **periodSeconds**.

7 Le nombre de fois où la sonde est autorisée à échouer. La valeur par défaut est de 3. Après le nombre de tentatives spécifié, le pod est marqué **Unready**.

8 Nombre de fois où la sonde doit signaler un succès, après un échec, pour être considérée comme réussie. La valeur par défaut est 1.

2. Créez la VMI en exécutant la commande suivante :

```
oc create -f <nom_du_fichier>.yaml
```

14.7.3. Définition d'une sonde de disponibilité TCP

Définissez une sonde de préparation TCP en définissant le champ **spec.readinessProbe.tcpSocket** de la configuration de l'instance de machine virtuelle (VMI).

Procédure

1. Inclure les détails de la sonde d'état de préparation TCP dans le fichier de configuration de l'IMV.

Exemple de sonde de disponibilité avec un test de socket TCP

```
...
spec:
  readinessProbe:
    initialDelaySeconds: 120 1
    periodSeconds: 20 2
    tcpSocket: 3
      port: 1500 4
    timeoutSeconds: 10 5
...
```

1 Temps, en secondes, écoulé entre le démarrage de l'IMV et le lancement de la sonde d'état de préparation.

2 Délai, en secondes, entre l'exécution des sondes. Le délai par défaut est de 10 secondes. Cette valeur doit être supérieure à **timeoutSeconds**.

3 L'action TCP à effectuer.

4 Le port de l'IMV que la sonde interroge.

5 Le nombre de secondes d'inactivité après lesquelles la sonde s'arrête et la VMI est considérée comme ayant échoué. La valeur par défaut est 1. Cette valeur doit être inférieure à **periodSeconds**.

2. Créez la VMI en exécutant la commande suivante :

```
oc create -f <nom_du_fichier>.yaml
```

14.7.4. Définition d'une sonde de vivacité HTTP

Définissez une sonde de disponibilité HTTP en définissant le champ **spec.livenessProbe.httpGet** de la configuration de l'instance de machine virtuelle (IMV). Vous pouvez définir des tests HTTP et TCP pour les sondes de validité de la même manière que pour les sondes de disponibilité. Cette procédure configure un exemple de sonde de disponibilité avec un test HTTP GET.

Procédure

1. Inclure les détails de la sonde HTTP dans le fichier de configuration de l'IMV.

Exemple de sonde de vivacité avec un test HTTP GET

```
# ...
spec:
  livenessProbe:
    initialDelaySeconds: 120 1
    periodSeconds: 20 2
    httpGet: 3
      port: 1500 4
      path: /healthz 5
      httpHeaders:
      - name: Custom-Header
        value: Awesome
    timeoutSeconds: 10 6
# ...
```

1 Délai, en secondes, entre le démarrage de l'IMV et le lancement de l'enquête sur l'état de conservation.

2 Délai, en secondes, entre l'exécution des sondes. Le délai par défaut est de 10 secondes. Cette valeur doit être supérieure à **timeoutSeconds**.

3 La requête HTTP GET à effectuer pour se connecter à l'IMV.

4 Le port de l'IMV que la sonde interroge. Dans l'exemple ci-dessus, la sonde interroge le port 1500. L'IMV installe et exécute un serveur HTTP minimal sur le port 1500 via cloud-init.

5 Le chemin d'accès au serveur HTTP. Dans l'exemple ci-dessus, si le gestionnaire du chemin d'accès au serveur **/healthz** renvoie un code de réussite, l'IMV est considérée comme saine. Si le gestionnaire renvoie un code d'échec, la VMI est supprimée et une nouvelle instance est créée.

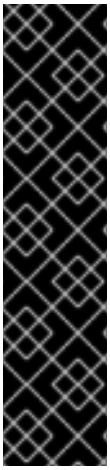
6 Le nombre de secondes d'inactivité après lesquelles la sonde s'arrête et la VMI est considérée comme ayant échoué. La valeur par défaut est 1. Cette valeur doit être inférieure à **periodSeconds**.

2. Créez la VMI en exécutant la commande suivante :

```
oc create -f <nom_du_fichier>.yaml
```

14.7.5. Définition d'une sonde ping de l'agent invité

Définissez une sonde ping de l'agent invité en définissant le champ **spec.readinessProbe.guestAgentPing** de la configuration de l'instance de machine virtuelle (VMI).



IMPORTANT

La sonde ping de l'agent invité est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas leur utilisation en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

Conditions préalables

- L'agent invité QEMU doit être installé et activé sur la machine virtuelle.

Procédure

1. Inclure les détails de la sonde ping de l'agent invité dans le fichier de configuration de l'IMV. Par exemple, les détails de la sonde ping de l'agent invité dans le fichier de configuration de l'IMV :

Exemple de sonde ping d'un agent invité

```
# ...
spec:
  readinessProbe:
    guestAgentPing: {} 1
    initialDelaySeconds: 120 2
    periodSeconds: 20 3
    timeoutSeconds: 10 4
    failureThreshold: 3 5
    successThreshold: 3 6
# ...
```

-

1. La sonde ping de l'agent invité pour se connecter à la VMI.
2. Facultatif : Le temps, en secondes, après le démarrage de l'IMV avant que la sonde de l'agent invité ne soit lancée.
3. Facultatif : Délai, en secondes, entre l'exécution des sondes. Le délai par défaut est de 10 secondes. Cette valeur doit être supérieure à **timeoutSeconds**.
4. Facultatif : Le nombre de secondes d'inactivité après lequel la sonde s'arrête et la VMI est considérée comme ayant échoué. La valeur par défaut est 1. Cette valeur doit être inférieure à **periodSeconds**.
5. Facultatif : Le nombre de fois où la sonde est autorisée à échouer. La valeur par défaut est 3. Après le nombre de tentatives spécifié, le pod est marqué **Unready**.
6. Facultatif : Nombre de fois où la sonde doit signaler un succès, après un échec, pour être considérée comme réussie. La valeur par défaut est 1.

2. Créez la VMI en exécutant la commande suivante :

```
oc create -f <nom_du_fichier>.yaml
```

14.7.6. Modèle : Fichier de configuration de la machine virtuelle pour la définition des contrôles de santé

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  labels:
    special: vm-fedora
  name: vm-fedora
spec:
  template:
    metadata:
      labels:
        special: vm-fedora
    spec:
      domain:
        devices:
          disks:
          - disk:
              bus: virtio
            name: containerdisk
          - disk:
              bus: virtio
            name: cloudinitdisk
        resources:
          requests:
            memory: 1024M
      readinessProbe:
        httpGet:
          port: 1500
        initialDelaySeconds: 120
```

```
      periodSeconds: 20
      timeoutSeconds: 10
      failureThreshold: 3
      successThreshold: 3
    terminationGracePeriodSeconds: 180
    volumes:
    - name: containerdisk
      containerDisk:
        image: kubevirt/fedora-cloud-registry-disk-demo
    - cloudInitNoCloud:
        userData: |-
          #cloud-config
          password: fedora
          chpasswd: { expire: False }
          bootcmd:
            - setenforce 0
            - dnf install -y nmap-ncat
            - systemd-run --unit=httpserver nc -klp 1500 -e '/usr/bin/echo -e HTTP/1.1 200 OK\\n\\nHello
World!'
      name: cloudinitdisk
```

14.7.7. Ressources supplémentaires

- Contrôler l'état de santé des applications à l'aide de bilans de santé

14.8. UTILISER LE TABLEAU DE BORD D'OPENSHIFT CONTAINER PLATFORM POUR OBTENIR DES INFORMATIONS SUR LES CLUSTERS

Accédez au tableau de bord d'OpenShift Container Platform, qui contient des informations de haut niveau sur le cluster, en cliquant sur **Home > Dashboards > Overview**dans la console web d'OpenShift Container Platform.

Le tableau de bord d'OpenShift Container Platform fournit diverses informations sur les clusters, capturées dans des tableaux de bord individuels *cards*.

14.8.1. A propos de la page des tableaux de bord d'OpenShift Container Platform

Le tableau de bord d'OpenShift Container Platform se compose des cartes suivantes :

- **Details** fournit un bref aperçu des détails de la grappe d'information.
  Les statuts comprennent **ok**, **error**, **warning**, **in progress**, et **unknown**. Les ressources peuvent ajouter des noms d'état personnalisés.
    - ID du groupe
    - Fournisseur
    - Version
- **Cluster Inventory** détaille le nombre de ressources et les statuts associés. Il est utile lorsqu'une intervention est nécessaire pour résoudre des problèmes, y compris des informations sur :
    - Nombre de nœuds
    - Nombre de gousses

- Demandes de volumes de stockage persistants
- Machines virtuelles (disponibles si OpenShift Virtualization est installé)
- Hôtes en métal nu dans le cluster, listés en fonction de leur état (disponible uniquement dans l'environnement **metal3**).

- **Cluster Health** résume l'état de santé actuel du cluster dans son ensemble, y compris les alertes et les descriptions pertinentes. Si OpenShift Virtualization est installé, la santé globale d'OpenShift Virtualization est également diagnostiquée. Si plusieurs sous-systèmes sont présents, cliquez sur **See All** pour afficher l'état de chaque sous-système.
- **Cluster Capacity** aident les administrateurs à comprendre quand des ressources supplémentaires sont nécessaires dans le cluster. Les graphiques contiennent un anneau intérieur qui affiche la consommation actuelle, tandis qu'un anneau extérieur affiche les seuils configurés pour la ressource, y compris des informations sur :
  - Temps CPU
  - Allocation de mémoire
  - Stockage consommé
  - Ressources réseau consommées
- **Cluster Utilization** montre la capacité de diverses ressources sur une période donnée, afin d'aider les administrateurs à comprendre l'ampleur et la fréquence d'une forte consommation de ressources.
- **Events** répertorie les messages liés à l'activité récente du cluster, comme la création d'un pod ou la migration d'une machine virtuelle vers un autre hôte.
- **Top Consumers** aide les administrateurs à comprendre comment les ressources de la grappe sont consommées. Cliquez sur une ressource pour accéder à une page détaillée répertoriant les pods et les nœuds qui consomment la plus grande quantité de la ressource de cluster spécifiée (CPU, mémoire ou stockage).

14.9. EXAMEN DE L'UTILISATION DES RESSOURCES PAR LES MACHINES VIRTUELLES

Les tableaux de bord de la console web d'OpenShift Container Platform fournissent des représentations visuelles des métriques du cluster pour vous aider à comprendre rapidement l'état de votre cluster. Les tableaux de bord font partie de la vue d'ensemble de la surveillance qui fournit une surveillance pour les composants de base de la plateforme.

Le tableau de bord OpenShift Virtualization fournit des données sur la consommation de ressources pour les machines virtuelles et les pods associés. Les mesures de visualisation affichées dans le tableau de bord OpenShift Virtualization sont basées sur des requêtes Prometheus Query Language (PromQL).

Un rôle de surveillance est nécessaire pour surveiller les espaces de noms définis par l'utilisateur dans le tableau de bord OpenShift Virtualization.

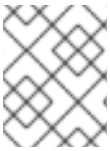
Vous pouvez voir l'utilisation des ressources pour une machine virtuelle spécifique sur la page**VirtualMachine details** → onglet **Metrics** dans la console web.

14.9.1. À propos de l'examen des principaux consommateurs

Dans le tableau de bord OpenShift Virtualization, vous pouvez sélectionner une période spécifique et afficher les principaux consommateurs de ressources au cours de cette période. Les principaux consommateurs sont les machines virtuelles ou les pods **virt-launcher** qui consomment le plus de ressources.

Le tableau suivant présente les ressources surveillées dans le tableau de bord et décrit les mesures associées à chaque ressource pour les principaux consommateurs.

Monitored resources	Description
Trafic d'échange de mémoire	Les machines virtuelles qui consomment le plus de mémoire lors de l'échange de mémoire.
attente vCPU	Machines virtuelles subissant le temps d'attente maximum (en secondes) pour leurs vCPU.
Utilisation de l'unité centrale par pod	Les pods **virt-launcher** qui utilisent le plus de CPU.
Trafic réseau	Machines virtuelles qui saturent le réseau en recevant le plus de trafic (en octets).
Trafic de stockage	Machines virtuelles dont le trafic lié au stockage est le plus important (en octets).
Stockage IOPS	Machines virtuelles ayant le plus grand nombre d'opérations d'E/S par seconde sur une période donnée.
Utilisation de la mémoire	Les pods **virt-launcher** qui utilisent le plus de mémoire (en octets).



NOTE

L'affichage de la consommation maximale de ressources est limité aux cinq premiers consommateurs.

14.9.2. Examen des principaux consommateurs

Dans la perspective **Administrator**, vous pouvez voir le tableau de bord OpenShift Virtualization où les principaux consommateurs de ressources sont affichés.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

1. Dans la perspective **Administrator** de la console web OpenShift Virtualization, naviguez vers **Observe → Dashboards**.

2. Sélectionnez le tableau de bord **KubeVirt/Infrastructure Resources/Top Consumers** dans la liste **Dashboard**.

3. Sélectionnez une période prédéfinie dans le menu déroulant pour **Period.** Vous pouvez consulter les données relatives aux principaux consommateurs dans les tableaux.

4. Facultatif : Cliquez sur **Inspect** pour afficher ou modifier la requête Prometheus Query Language (PromQL) associée aux principaux consommateurs d'une table.

14.9.3. Ressources supplémentaires

- Aperçu de la surveillance
- Examen des tableaux de bord de suivi

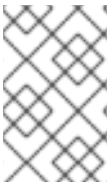
14.10. SURVEILLANCE, JOURNALISATION ET TÉLÉMÉTRIE DES CLUSTERS OPENSHIFT CONTAINER PLATFORM

OpenShift Container Platform fournit diverses ressources pour la surveillance au niveau du cluster.

14.10.1. À propos de la surveillance de la plateforme OpenShift Container

OpenShift Container Platform comprend une pile de surveillance préconfigurée, préinstallée et auto-actualisée qui fournit **monitoring for core platform components**. OpenShift Container Platform offre les meilleures pratiques de surveillance dès sa sortie de l'emballage. Un ensemble d'alertes est inclus par défaut et permet d'informer immédiatement les administrateurs de clusters des problèmes rencontrés par ces derniers. Les tableaux de bord par défaut de la console web d'OpenShift Container Platform incluent des représentations visuelles des métriques du cluster pour vous aider à comprendre rapidement l'état de votre cluster.

Après l'installation d'OpenShift Container Platform 4.12, les administrateurs de clusters peuvent optionnellement activer **monitoring for user-defined projects**. En utilisant cette fonctionnalité, les administrateurs de clusters, les développeurs et les autres utilisateurs peuvent spécifier comment les services et les pods sont surveillés dans leurs propres projets. Vous pouvez ensuite interroger les métriques, consulter les tableaux de bord et gérer les règles d'alerte et les silences pour vos propres projets dans la console web d'OpenShift Container Platform.



NOTE

Les administrateurs de clusters peuvent accorder aux développeurs et à d'autres utilisateurs l'autorisation de surveiller leurs propres projets. Les privilèges sont accordés en attribuant l'un des rôles de surveillance prédéfinis.

14.10.2. À propos des composants du sous-système de journalisation

Les composants du sous-système de journalisation comprennent un collecteur déployé sur chaque nœud du cluster OpenShift Container Platform qui collecte tous les journaux des nœuds et des conteneurs et les écrit dans un magasin de journaux. Vous pouvez utiliser une interface web centralisée pour créer des visualisations et des tableaux de bord riches avec les données agrégées.

Les principaux composants du sous-système de journalisation sont les suivants :

- collection - C'est le composant qui collecte les logs du cluster, les formate et les transmet au magasin de logs. L'implémentation actuelle est Fluentd.

- log store - C'est l'endroit où sont stockés les journaux. L'implémentation par défaut est Elasticsearch. Vous pouvez utiliser le magasin de journaux Elasticsearch par défaut ou transférer les journaux vers des magasins de journaux externes. Le magasin de journaux par défaut est optimisé et testé pour le stockage à court terme.

- visualisation - Il s'agit du composant de l'interface utilisateur que vous pouvez utiliser pour afficher les journaux, les graphiques, les diagrammes, etc. L'implémentation actuelle est Kibana.

Pour plus d'informations sur OpenShift Logging, voir la documentation OpenShift Logging.

14.10.3. À propos de la télémétrie

Telemetry envoie à Red Hat un sous-ensemble soigneusement choisi de métriques de surveillance du cluster. Le client Telemeter récupère les valeurs des métriques toutes les quatre minutes et trente secondes et télécharge les données vers Red Hat. Ces mesures sont décrites dans ce document.

Ce flux de données est utilisé par Red Hat pour surveiller les clusters en temps réel et pour réagir si nécessaire aux problèmes qui ont un impact sur nos clients. Il permet également à Red Hat de déployer les mises à niveau d'OpenShift Container Platform auprès des clients afin de minimiser l'impact sur le service et d'améliorer continuellement l'expérience de mise à niveau.

Ces informations de débogage sont disponibles pour les équipes d'assistance et d'ingénierie de Red Hat avec les mêmes restrictions que l'accès aux données signalées par les cas d'assistance. Toutes les informations sur les clusters connectés sont utilisées par Red Hat pour améliorer OpenShift Container Platform et rendre son utilisation plus intuitive.

14.10.3.1. Informations collectées par la télémétrie

Les informations suivantes sont collectées par Telemetry :

14.10.3.1.1. Informations sur le système

- Informations sur la version, y compris la version du cluster OpenShift Container Platform et les détails de la mise à jour installée qui sont utilisés pour déterminer la disponibilité de la version de la mise à jour

- Informations sur les mises à jour, y compris le nombre de mises à jour disponibles par cluster, le canal et le référentiel d'images utilisés pour une mise à jour, les informations sur la progression de la mise à jour et le nombre d'erreurs survenues lors d'une mise à jour

- L'identifiant aléatoire unique qui est généré lors d'une installation

- Détails de configuration qui aident le support Red Hat à fournir une assistance bénéfique aux clients, y compris la configuration des nœuds au niveau de l'infrastructure cloud, les noms d'hôtes, les adresses IP, les noms de pods Kubernetes, les espaces de noms et les services

- Les composants du framework OpenShift Container Platform installés dans un cluster, ainsi que leur état et leur statut

- Événements pour tous les espaces de noms répertoriés comme "objets apparentés" pour un opérateur dégradé

- Informations sur les logiciels dégradés

- Informations sur la validité des certificats

- Le nom de la plateforme du fournisseur sur laquelle OpenShift Container Platform est déployée et l'emplacement du centre de données

14.10.3.1.2. Information sur la taille

- Informations sur le dimensionnement des grappes, des types de machines et des machines, y compris le nombre de cœurs de CPU et la quantité de RAM utilisée pour chacun d'entre eux
- Nombre d'instances de machines virtuelles en cours d'exécution dans un cluster
- Le nombre de membres etcd et le nombre d'objets stockés dans le cluster etcd
- Nombre d'applications construites par type de stratégie de construction

14.10.3.1.3. Informations sur l'utilisation

- Informations sur l'utilisation des composants, des fonctionnalités et des extensions
- Détails d'utilisation des aperçus technologiques et des configurations non prises en charge

La télémétrie ne collecte pas d'informations d'identification telles que les noms d'utilisateur ou les mots de passe. Le Chapeau Rouge n'a pas l'intention de collecter des informations personnelles. Si Red Hat découvre que des informations personnelles ont été reçues par inadvertance, Red Hat supprimera ces informations. Dans la mesure où les données de télémétrie constituent des données personnelles, veuillez vous référer à la Déclaration de confidentialité de Red Hat pour plus d'informations sur les pratiques de confidentialité de Red Hat.

14.10.4. Commandes de dépannage et de débogage du CLI

Pour obtenir une liste des commandes de dépannage et de débogage du client **oc**, consultez la documentation des outils CLI de OpenShift Container Platform.

14.11. EXÉCUTION DES VÉRIFICATIONS DE LA GRAPPE

OpenShift Virtualization inclut des checkups prédéfinis qui peuvent être utilisés pour la maintenance et le dépannage des clusters.

IMPORTANT

Le cadre de vérification des clusters d'OpenShift Container Platform est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

14.11.1. À propos du cadre de vérification des clusters d'OpenShift Container Platform

Un *checkup* est une charge de travail de test automatisée qui vous permet de vérifier si une fonctionnalité spécifique du cluster fonctionne comme prévu. Le cadre de vérification du cluster utilise les ressources natives de Kubernetes pour configurer et exécuter la vérification.

En utilisant des vérifications prédéfinies, les administrateurs et les développeurs de clusters peuvent améliorer la maintenabilité des clusters, dépanner les comportements inattendus, minimiser les erreurs et gagner du temps. Ils peuvent également examiner les résultats de la vérification et les partager avec des experts pour une analyse plus approfondie. Les fournisseurs peuvent rédiger et publier des checkups pour les fonctionnalités ou les services qu'ils proposent et vérifier que les environnements de leurs clients sont configurés correctement.

L'exécution d'un contrôle prédéfini dans un espace de noms existant implique la configuration d'un compte de service pour le contrôle, la création des objets **Role** et **RoleBinding** pour le compte de service, l'activation des autorisations pour le contrôle et la création de la carte de configuration d'entrée et de la tâche de contrôle. Vous pouvez exécuter un contrôle plusieurs fois.

IMPORTANT

Vous devez toujours :

- Vérifiez que l'image de contrôle provient d'une source fiable avant de l'appliquer.
- Vérifiez les autorisations de contrôle avant de créer les objets **Role** et **RoleBinding**.

14.11.2. Vérification de la connectivité et de la latence du réseau pour les machines virtuelles sur un réseau secondaire

Vous utilisez un contrôle prédéfini pour vérifier la connectivité réseau et mesurer la latence entre deux machines virtuelles (VM) connectées à une interface réseau secondaire.

Pour effectuer un contrôle pour la première fois, suivez les étapes de la procédure.

Si vous avez déjà effectué un contrôle, passez à l'étape 5 de la procédure, car les étapes d'installation du cadre et d'activation des autorisations pour le contrôle ne sont pas nécessaires.

Conditions préalables

- You installed the OpenShift CLI (**oc**).
- La grappe compte au moins deux nœuds de travail.
- Le plugin Multus Container Network Interface (CNI) est installé sur le cluster.
- Vous avez configuré une définition d'attachement réseau pour un espace de noms.

Procédure

1. Créez un fichier manifeste contenant les objets **ServiceAccount**, **Role** et **RoleBinding** avec les autorisations requises par le checkup pour l'accès au cluster :

Exemple 14.3. Exemple de fichier manifeste des rôles

```
---
apiVersion: v1
kind: ServiceAccount
```

```
  metadata:
    name: vm-latency-checkup-sa
  ---
  apiVersion: rbac.authorization.k8s.io/v1
  kind: Role
  metadata:
    name: kubevirt-vm-latency-checker
  rules:
  - apiGroups: ["kubevirt.io"]
    resources: ["virtualmachineinstances"]
    verbs: ["get", "create", "delete"]
  - apiGroups: ["subresources.kubevirt.io"]
    resources: ["virtualmachineinstances/console"]
    verbs: ["get"]
  - apiGroups: ["k8s.cni.cncf.io"]
    resources: ["network-attachment-definitions"]
    verbs: ["get"]
  ---
  apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
  metadata:
    name: kubevirt-vm-latency-checker
  subjects:
  - kind: ServiceAccount
    name: vm-latency-checkup-sa
  roleRef:
    kind: Role
    name: kubevirt-vm-latency-checker
    apiGroup: rbac.authorization.k8s.io
  ---
  apiVersion: rbac.authorization.k8s.io/v1
  kind: Role
  metadata:
    name: kiagnose-configmap-access
  rules:
  - apiGroups: [ "" ]
    resources: [ "configmaps" ]
    verbs: ["get", "update"]
  ---
  apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
  metadata:
    name: kiagnose-configmap-access
  subjects:
  - kind: ServiceAccount
    name: vm-latency-checkup-sa
  roleRef:
    kind: Role
    name: kiagnose-configmap-access
    apiGroup: rbac.authorization.k8s.io
```

2. Appliquer le manifeste des rôles de contrôle :

```
oc apply -n <target_namespace> -f <latency_roles>.yaml 1
```

1. **<target_namespace>** est l'espace de noms dans lequel le contrôle doit être exécuté. Il doit s'agir d'un espace de noms existant où réside l'objet **NetworkAttachmentDefinition**.

3. Créer un manifeste **ConfigMap** qui contient les paramètres d'entrée du contrôle. La carte de configuration fournit au cadre de travail les données d'entrée nécessaires à l'exécution du contrôle et stocke également les résultats de ce dernier.

   Exemple de carte de configuration d'entrée

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: kubevirt-vm-latency-checkup-config
data:
  spec.timeout: 5m
  spec.param.network_attachment_definition_namespace: <target_namespace>
  spec.param.network_attachment_definition_name: "blue-network" (1)
  spec.param.max_desired_latency_milliseconds: "10" (2)
  spec.param.sample_duration_seconds: "5" (3)
  spec.param.source_node: "worker1" (4)
  spec.param.target_node: "worker2" (5)
```

1. Le nom de l'objet **NetworkAttachmentDefinition**.
2. Facultatif : La latence maximale souhaitée, en millisecondes, entre les machines virtuelles. Si la latence mesurée dépasse cette valeur, le contrôle échoue.
3. Facultatif : La durée de la vérification de la latence, en secondes.
4. Facultatif : Lorsqu'il est spécifié, le temps de latence est mesuré entre ce nœud et le nœud cible. Si le nœud source est spécifié, le champ **spec.param.target_node** ne peut pas être vide.
5. Facultatif : Lorsqu'elle est spécifiée, la latence est mesurée entre le nœud source et ce nœud.

4. Appliquer le manifeste de la carte de configuration dans l'espace de noms cible :

```
oc apply -n <target_namespace> -f <latency_config_map>.yaml
```

5. Créez un objet **Job** pour exécuter le contrôle :

   Exemple de manifeste d'emploi

```
apiVersion: batch/v1
kind: Job
metadata:
  name: kubevirt-vm-latency-checkup
spec:
  backoffLimit: 0
  template:
    spec:
      serviceAccountName: vm-latency-checkup-sa
```

```
      restartPolicy: Never
      containers:
        - name: vm-latency-checkup
          image: registry.redhat.io/container-native-virtualization/vm-network-latency-
checkup:v4.12.0
          securityContext:
            allowPrivilegeEscalation: false
            capabilities:
              drop: ["ALL"]
            runAsNonRoot: true
            seccompProfile:
              type: "RuntimeDefault"
          env:
            - name: CONFIGMAP_NAMESPACE
              value: <target_namespace>
            - name: CONFIGMAP_NAME
              value: kubevirt-vm-latency-checkup-config
```

6. Appliquez le manifeste **Job**. Le contrôle utilise l'utilitaire ping pour vérifier la connectivité et mesurer la latence.

```
oc apply -n <target_namespace> -f <latency_job>.yaml
```

7. Attendez que le travail soit terminé :

```
$ oc wait job kubevirt-vm-latency-checkup -n <target_namespace> --for condition=complete -
-timeout 6m
```

8. Examinez les résultats du contrôle de latence en exécutant la commande suivante. Si la latence maximale mesurée est supérieure à la valeur de l'attribut **spec.param.max_desired_latency_milliseconds**, le contrôle échoue et renvoie une erreur.

```
$ oc get configmap kubevirt-vm-latency-checkup-config -n <target_namespace> -o yaml
```

Exemple de sortie config map (succès)

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: kubevirt-vm-latency-checkup-config
  namespace: <target_namespace>
data:
  spec.timeout: 5m
  spec.param.network_attachment_definition_namespace: <target_namespace>
  spec.param.network_attachment_definition_name: "blue-network"
  spec.param.max_desired_latency_milliseconds: "10"
  spec.param.sample_duration_seconds: "5"
  spec.param.source_node: "worker1"
  spec.param.target_node: "worker2"
  status.succeeded: "true"
  status.failureReason: ""
  status.completionTimestamp: "2022-01-01T09:00:00Z"
  status.startTimestamp: "2022-01-01T09:00:07Z"
  status.result.avgLatencyNanoSec: "177000"
```

```
status.result.maxLatencyNanoSec: "244000" (1)
status.result.measurementDurationSec: "5"
status.result.minLatencyNanoSec: "135000"
status.result.sourceNode: "worker1"
status.result.targetNode: "worker2"
```

(1) La latence maximale mesurée en nanosecondes.

9. Facultatif : Pour afficher le journal détaillé du travail en cas d'échec du contrôle, utilisez la commande suivante :

```
oc logs job.batch/kubevirt-vm-latency-checkup -n <target_namespace>
```

10. Supprimez les ressources job et config map que vous avez créées précédemment en exécutant les commandes suivantes :

```
oc delete job -n <target_namespace> kubevirt-vm-latency-checkup
```

```
oc delete config-map -n <target_namespace> kubevirt-vm-latency-checkup-config
```

11. Facultatif : si vous ne prévoyez pas d'effectuer un autre contrôle, supprimez le rôle de contrôle et les fichiers manifestes du cadre.

```
oc delete -f <nom_du_fichier>.yaml
```

14.11.3. Ressources supplémentaires

- Attacher une machine virtuelle à plusieurs réseaux

14.12. REQUÊTES PROMETHEUS POUR LES RESSOURCES VIRTUELLES

OpenShift Virtualization fournit des métriques que vous pouvez utiliser pour surveiller la consommation des ressources de l'infrastructure du cluster, y compris les vCPU, le réseau, le stockage et le swapping de la mémoire invitée. Vous pouvez également utiliser les métriques pour interroger l'état de la migration en direct.

Utilisez le tableau de bord de surveillance d'OpenShift Container Platform pour interroger les métriques de virtualisation.

14.12.1. Conditions préalables

- Pour utiliser la métrique vCPU, l'argument noyau **schedstats=enable** doit être appliqué à l'objet **MachineConfig**. Cet argument du noyau permet d'activer les statistiques du planificateur utilisées pour le débogage et le réglage des performances et ajoute une charge supplémentaire mineure au planificateur. Voir la documentation sur les tâches de configuration de la machine OpenShift Container Platform pour plus d'informations sur l'application d'un argument de noyau.

- Pour que les requêtes de permutation de la mémoire des invités renvoient des données, la permutation de la mémoire doit être activée sur les invités virtuels.

14.12.2. A propos de l'interrogation des métriques

Le tableau de bord de surveillance d'OpenShift Container Platform vous permet d'exécuter des requêtes Prometheus Query Language (PromQL) pour examiner les mesures visualisées sur un graphique. Cette fonctionnalité fournit des informations sur l'état d'un cluster et de toute charge de travail définie par l'utilisateur que vous surveillez.

En tant que **cluster administrator**, vous pouvez interroger les métriques pour tous les projets principaux d'OpenShift Container Platform et les projets définis par l'utilisateur.

En tant que **developer**, vous devez spécifier un nom de projet lorsque vous interrogez les métriques. Vous devez disposer des privilèges requis pour afficher les métriques du projet sélectionné.

14.12.2.1. Interroger les métriques de tous les projets en tant qu'administrateur de cluster

En tant qu'administrateur de cluster ou en tant qu'utilisateur disposant de permissions de visualisation pour tous les projets, vous pouvez accéder aux métriques pour tous les projets par défaut d'OpenShift Container Platform et les projets définis par l'utilisateur dans l'interface utilisateur des métriques.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur avec le rôle **cluster-admin** ou avec des permissions de visualisation pour tous les projets.
- Vous avez installé l'OpenShift CLI (**oc**).

Procédure

1. Sélectionnez la perspective **Administrator** dans la console web de OpenShift Container Platform.
2. Sélectionnez **Observe → Metrics**.
3. Sélectionnez **Insert Metric at Cursor** pour afficher une liste de requêtes prédéfinies.
4. Pour créer une requête personnalisée, ajoutez votre requête Prometheus Query Language (PromQL) au champ **Expression**.

NOTE

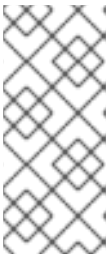
Lorsque vous saisissez une expression PromQL, des suggestions d'autocomplétion apparaissent dans une liste déroulante. Ces suggestions incluent des fonctions, des métriques, des étiquettes et des jetons de temps. Vous pouvez utiliser les flèches du clavier pour sélectionner l'un des éléments suggérés, puis appuyer sur Entrée pour l'ajouter à votre expression. Vous pouvez également déplacer le pointeur de votre souris sur un élément suggéré pour afficher une brève description de cet élément.

5. Pour ajouter plusieurs requêtes, sélectionnez **Add Query**.
6. Pour dupliquer une requête existante, sélectionnez  à côté de la requête, puis choisissez **Duplicate query**.

7. Pour supprimer une requête, sélectionnez à côté de la requête, puis choisissez **Delete query**.

8. Pour empêcher l'exécution d'une requête, sélectionnez à côté de la requête et choisissez **Disable query**.

9. Pour exécuter les requêtes que vous avez créées, sélectionnez **Run Queries**. Les métriques des requêtes sont visualisées sur le graphique. Si une requête n'est pas valide, l'interface utilisateur affiche un message d'erreur.



NOTE

Les requêtes qui portent sur de grandes quantités de données peuvent dépasser le temps imparti ou surcharger le navigateur lors de l'affichage de graphiques de séries temporelles. Pour éviter cela, sélectionnez **Hide graph** et calibrez votre requête en utilisant uniquement le tableau des métriques. Ensuite, après avoir trouvé une requête réalisable, activez le tracé pour dessiner les graphiques.

10. Facultatif : L'URL de la page contient maintenant les requêtes que vous avez exécutées. Pour réutiliser cet ensemble de requêtes à l'avenir, enregistrez cette URL.

14.12.2.2. Interroger les métriques pour des projets définis par l'utilisateur en tant que développeur

Vous pouvez accéder aux métriques d'un projet défini par l'utilisateur en tant que développeur ou en tant qu'utilisateur disposant d'autorisations de visualisation du projet.

Dans la perspective **Developer**, l'interface utilisateur Metrics comprend des requêtes prédéfinies sur l'unité centrale, la mémoire, la bande passante et les paquets réseau pour le projet sélectionné. Vous pouvez également exécuter des requêtes Prometheus Query Language (PromQL) personnalisées pour l'unité centrale, la mémoire, la bande passante, les paquets réseau et les métriques d'application pour le projet.

NOTE

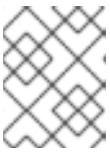
Les développeurs ne peuvent utiliser que la perspective **Developer** et non la perspective **Administrator**. En tant que développeur, vous ne pouvez interroger les métriques que pour un seul projet à la fois dans la page **Observe** -→ **Metrics** de la console web pour votre projet défini par l'utilisateur.

Conditions préalables

- Vous avez accès au cluster en tant que développeur ou en tant qu'utilisateur disposant d'autorisations de visualisation pour le projet dont vous consultez les métriques.
- Vous avez activé la surveillance pour les projets définis par l'utilisateur.
- Vous avez déployé un service dans un projet défini par l'utilisateur.
- Vous avez créé une définition de ressource personnalisée (CRD) **ServiceMonitor** pour le service afin de définir la manière dont le service est surveillé.

Procédure

1. Sélectionnez la perspective **Developer** dans la console web de OpenShift Container Platform.
2. Sélectionnez **Observe → Metrics**.
3. Dans la liste **Project:**, sélectionnez le projet pour lequel vous souhaitez afficher les mesures.
4. Sélectionnez une requête dans la liste **Select query** ou créez une requête PromQL personnalisée basée sur la requête sélectionnée en sélectionnant **Show PromQL**.
5. Optionnel : Sélectionnez **Custom query** dans la liste **Select query** pour saisir une nouvelle requête. Au fur et à mesure de la saisie, des suggestions d'autocomplétion apparaissent dans une liste déroulante. Ces suggestions comprennent des fonctions et des mesures. Cliquez sur un élément suggéré pour le sélectionner.

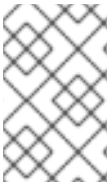


NOTE

Dans la perspective **Developer**, vous ne pouvez exécuter qu'une seule requête à la fois.

14.12.3. Mesures de virtualisation

Les descriptions des mesures suivantes comprennent des exemples de requêtes en Prometheus Query Language (PromQL). Ces mesures ne constituent pas une API et peuvent changer d'une version à l'autre.



NOTE

Les exemples suivants utilisent les requêtes **topk** qui spécifient une période de temps. Si des machines virtuelles sont supprimées au cours de cette période, elles peuvent toujours apparaître dans le résultat de la requête.

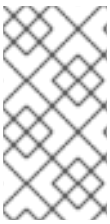
14.12.3.1. métriques vCPU

La requête suivante permet d'identifier les machines virtuelles qui attendent une entrée/sortie (E/S) :

kubevirt_vmi_vcpu_wait_seconds

Renvoie le temps d'attente (en secondes) pour le vCPU d'une machine virtuelle. Type : Compteur.

Une valeur supérieure à '0' signifie que la vCPU veut s'exécuter, mais que le planificateur de l'hôte ne peut pas encore le faire. Cette impossibilité d'exécution indique qu'il y a un problème au niveau des entrées/sorties.



NOTE

Pour interroger la métrique vCPU, l'argument de noyau **schedstats=enable** doit d'abord être appliqué à l'objet **MachineConfig**. Cet argument du noyau permet d'activer les statistiques de l'ordonnanceur utilisées pour le débogage et le réglage des performances et ajoute une charge supplémentaire mineure à l'ordonnanceur.

Exemple de requête sur le temps d'attente vCPU

```
topk(3, sum by (name, namespace) (rate(kubevirt_vmi_vcpu_wait_seconds[6m]))) > 0 1
```

-

1 Cette requête renvoie les 3 premières machines virtuelles en attente d'E/S à chaque instant sur une période de six minutes.

14.12.3.2. Mesures du réseau

Les requêtes suivantes permettent d'identifier les machines virtuelles qui saturent le réseau :

kubevirt_vmi_network_receive_bytes_total

Renvoie la quantité totale de trafic reçu (en octets) sur le réseau de la machine virtuelle. Type : Compteur.

kubevirt_vmi_network_transmit_bytes_total

Renvoie la quantité totale de trafic transmise (en octets) sur le réseau de la machine virtuelle. Type : Compteur.

Exemple d'interrogation sur le trafic réseau

```
topk(3, sum by (name, namespace) (rate(kubevirt_vmi_network_receive_bytes_total[6m])) sum by (name, namespace) (rate(kubevirt_vmi_network_transmit_bytes_total[6m]))) > 0 1
```

1 Cette requête renvoie les trois machines virtuelles qui transmettent le plus de trafic réseau à chaque instant sur une période de six minutes.

14.12.3.3. Mesures de stockage

14.12.3.3.1. Trafic lié au stockage

Les requêtes suivantes permettent d'identifier les machines virtuelles qui écrivent de grandes quantités de données :

kubevirt_vmi_storage_read_traffic_bytes_total

Renvoie la quantité totale (en octets) du trafic lié au stockage de la machine virtuelle. Type : Compteur.

kubevirt_vmi_storage_write_traffic_bytes_total

Renvoie la quantité totale d'écritures de stockage (en octets) du trafic lié au stockage de la machine virtuelle. Type : Compteur.

Exemple d'interrogation sur le trafic lié au stockage

```
topk(3, sum by (name, namespace) (rate(kubevirt_vmi_storage_read_traffic_bytes_total[6m])) sum by (name, namespace) (rate(kubevirt_vmi_storage_write_traffic_bytes_total[6m]))) > 0 1
```

1 Cette requête renvoie les 3 machines virtuelles qui effectuent le plus de trafic de stockage à chaque instant sur une période de six minutes.

14.12.3.3.2. Données de l'instantané de stockage

kubevirt_vmsnapshot_disks_restored_from_source_total

Renvoie le nombre total de disques de machine virtuelle restaurés à partir de la machine virtuelle source. Type : Jauge.

kubevirt_vmsnapshot_disks_restored_from_source_bytes

Renvoie la quantité d'espace en octets restaurée à partir de la machine virtuelle source. Type : Jauge.

Exemples de requêtes de données d'instantanés de stockage

```
kubevirt_vmsnapshot_disks_restored_from_source_total{vm_name="simple-vm",
vm_namespace="default"} 1
```

1 Cette requête renvoie le nombre total de disques de la machine virtuelle restaurés à partir de la machine virtuelle source.

```
kubevirt_vmsnapshot_disks_restored_from_source_bytes{vm_name="simple-vm",
vm_namespace="default"} 1
```

1 Cette requête renvoie la quantité d'espace en octets restaurée à partir de la machine virtuelle source.

14.12.3.3.3. Performance des E/S

Les requêtes suivantes permettent de déterminer les performances d'E/S des périphériques de stockage :

kubevirt_vmi_storage_iops_read_total

Renvoie la quantité d'opérations d'E/S en écriture que la machine virtuelle effectue par seconde. Type : Compteur.

kubevirt_vmi_storage_iops_write_total

Renvoie la quantité d'opérations d'E/S en lecture que la machine virtuelle effectue par seconde. Type : Compteur.

Exemple de requête sur les performances des E/S

```
topk(3, sum by (name, namespace) (rate(kubevirt_vmi_storage_iops_read_total[6m])) sum by (name,
namespace) (rate(kubevirt_vmi_storage_iops_write_total[6m]))) > 0 1
```

1 Cette requête renvoie les 3 machines virtuelles effectuant le plus grand nombre d'opérations d'E/S par seconde à chaque instant sur une période de six minutes.

14.12.3.4. Mesures de permutation de la mémoire invitée

Les requêtes suivantes permettent d'identifier les invités autorisés à permuter qui effectuent le plus de permutations de mémoire :

kubevirt_vmi_memory_swap_in_traffic_bytes_total

Renvoie la quantité totale (en octets) de mémoire que l'invité virtuel est en train d'échanger. Type : Jauge.

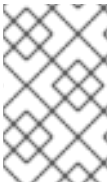
kubevirt_vmi_memory_swap_out_traffic_bytes_total

Renvoie la quantité totale (en octets) de mémoire que l'invité virtuel est en train d'échanger. Type : Jauge.

Exemple de requête de permutation de mémoire

```
topk(3, sum by (name, namespace) (rate(kubevirt_vmi_memory_swap_in_traffic_bytes_total[6m])) sum by (name, namespace) (rate(kubevirt_vmi_memory_swap_out_traffic_bytes_total[6m]))) > 0 1
```

1 Cette requête renvoie les 3 principales machines virtuelles pour lesquelles l'invité effectue le plus de permutations de mémoire à chaque instant sur une période de six minutes.



NOTE

La permutation de la mémoire indique que la machine virtuelle est soumise à une pression de mémoire. L'augmentation de l'allocation de mémoire de la machine virtuelle peut atténuer ce problème.

14.12.4. Mesures de migration en temps réel

Les paramètres suivants peuvent être interrogés pour connaître l'état de la migration en temps réel :

kubevirt_migrate_vmi_data_processed_bytes

La quantité de données du système d'exploitation invité (OS) qui a migré vers la nouvelle machine virtuelle (VM). Type : Jauge.

kubevirt_migrate_vmi_data_remaining_bytes

Quantité de données du système d'exploitation invité restant à migrer. Type : Jauge.

kubevirt_migrate_vmi_dirty_memory_rate_bytes

La vitesse à laquelle la mémoire devient sale dans le système d'exploitation invité. La mémoire sale est constituée de données qui ont été modifiées mais qui n'ont pas encore été écrites sur le disque. Type : Jauge.

kubevirt_migrate_vmi_pending_count

Le nombre de migrations en attente. Type : Jauge.

kubevirt_migrate_vmi_scheduling_count

Le nombre de migrations d'ordonnancement. Type : Jauge.

kubevirt_migrate_vmi_running_count

Le nombre de migrations en cours. Type : Jauge.

kubevirt_migrate_vmi_succeeded

Le nombre de migrations effectuées avec succès. Type : Jauge.

kubevirt_migrate_vmi_failed

Le nombre de migrations qui ont échoué. Type : Jauge.

14.12.5. Ressources supplémentaires

- Aperçu de la surveillance
- Interroger Prométhée
- Exemples de requêtes Prometheus

14.13. EXPOSITION DE MESURES PERSONNALISÉES POUR LES MACHINES VIRTUELLES

OpenShift Container Platform comprend une pile de surveillance préconfigurée, préinstallée et auto-actualisée qui assure la surveillance des composants de base de la plate-forme. Cette pile de surveillance est basée sur le système de surveillance Prometheus. Prometheus est une base de données de séries temporelles et un moteur d'évaluation de règles pour les métriques.

En plus d'utiliser la pile de surveillance d'OpenShift Container Platform, vous pouvez activer la surveillance pour des projets définis par l'utilisateur à l'aide de la CLI et demander des mesures personnalisées qui sont exposées pour les machines virtuelles par le biais du service **node-exporter**.

14.13.1. Configuration du service d'exportation de nœuds

L'agent node-exporter est déployé sur chaque machine virtuelle du cluster à partir de laquelle vous souhaitez collecter des métriques. Configurez l'agent node-exporter en tant que service pour exposer les mesures et processus internes associés aux machines virtuelles.

Conditions préalables

- Install the OpenShift Container Platform CLI **oc**.
- Connectez-vous au cluster en tant qu'utilisateur disposant des privilèges **cluster-admin**.
- Créez l'objet **cluster-monitoring-config ConfigMap** dans le projet **openshift-monitoring**.
- Configurez l'objet **user-workload-monitoring-config ConfigMap** dans le projet **openshift-user-workload-monitoring** en réglant **enableUserWorkload** sur **true**.

Procédure

1. Créez le fichier YAML **Service**. Dans l'exemple suivant, le fichier s'appelle **node-exporter-service.yaml**.

```
kind: Service
apiVersion: v1
metadata:
  name: node-exporter-service 1
  namespace: dynamation 2
  labels:
    servicetype: metrics 3
spec:
  ports:
    - name: exmet 4
      protocol: TCP
      port: 9100 5
      targetPort: 9100 6
  type: ClusterIP
  selector:
    monitor: metrics 7
```

1. Le service d'exportation de nœuds qui expose les métriques des machines virtuelles.
2. L'espace de noms dans lequel le service est créé.

3. L'étiquette du service. Le site **ServiceMonitor** utilise ce label pour faire correspondre ce service.
4. Le nom donné au port qui expose les métriques sur le port 9100 pour le service **ClusterIP**.
5. Le port cible utilisé par **node-exporter-service** pour écouter les demandes.
6. Le numéro de port TCP de la machine virtuelle configurée avec l'étiquette **monitor**.
7. L'étiquette utilisée pour faire correspondre les pods de la machine virtuelle. Dans cet exemple, tout module de machine virtuelle portant l'étiquette **monitor** et la valeur **metrics** sera pris en compte.

2. Créer le service node-exporter :

```
$ oc create -f node-exporter-service.yaml
```

14.13.2. Configuration d'une machine virtuelle avec le service d'exportation de nœuds

Téléchargez le fichier **node-exporter** sur la machine virtuelle. Ensuite, créez un service **systemd** qui exécute le service node-exporter lorsque la machine virtuelle démarre.

Conditions préalables

- Les pods du composant sont en cours d'exécution dans le projet **openshift-user-workload-monitoring**.
- Attribuez le rôle **monitoring-edit** aux utilisateurs qui doivent surveiller ce projet défini par l'utilisateur.

Procédure

1. Connectez-vous à la machine virtuelle.
2. Téléchargez le fichier **node-exporter** sur la machine virtuelle en utilisant le chemin d'accès au répertoire qui s'applique à la version du fichier **node-exporter**.

```
$ wget
https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-
1.3.1.linux-amd64.tar.gz
```

3. Extraire l'exécutable et le placer dans le répertoire **/usr/bin**.

```
$ sudo tar xvf node_exporter-1.3.1.linux-amd64.tar.gz \
    --directory /usr/bin --strip 1 "*/node_exporter"
```

4. Créez un fichier **node_exporter.service** dans le répertoire suivant : **/etc/systemd/system**. Ce fichier de service **systemd** exécute le service node-exporter lorsque la machine virtuelle redémarre.

```
[Unit]
Description=Prometheus Metrics Exporter
```

```
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=root
ExecStart=/usr/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

5. Activez et démarrez le service **systemd**.

```
$ sudo systemctl enable node_exporter.service
$ sudo systemctl start node_exporter.service
```

Vérification

- Vérifiez que l'agent node-exporter rapporte les métriques de la machine virtuelle.

```
$ curl http://localhost:9100/metrics
```

Exemple de sortie

```
go_gc_duration_seconds{quantile="0"} 1.5244e-05
go_gc_duration_seconds{quantile="0.25"} 3.0449e-05
go_gc_duration_seconds{quantile="0.5"} 3.7913e-05
```

14.13.3. Création d'une étiquette de surveillance personnalisée pour les machines virtuelles

Pour permettre l'interrogation de plusieurs machines virtuelles à partir d'un seul service, ajoutez une étiquette personnalisée dans le fichier YAML de la machine virtuelle.

Conditions préalables

- Install the OpenShift Container Platform CLI **oc**.
- Connectez-vous en tant qu'utilisateur disposant des privilèges **cluster-admin**.
- Accès à la console web pour arrêter et redémarrer une machine virtuelle.

Procédure

1. Modifiez la spécification **template** de votre fichier de configuration de la machine virtuelle. Dans cet exemple, l'étiquette **monitor** a la valeur **metrics**.

```
spec:
  template:
    metadata:
```

```
    labels:
      monitor: metrics
```

2. Arrêtez et redémarrez la machine virtuelle pour créer un nouveau pod avec le nom d'étiquette donné à l'étiquette **monitor**.

14.13.3.1. Interroger le service node-exporter pour obtenir des métriques

Les mesures sont exposées pour les machines virtuelles par le biais d'un point de terminaison de service HTTP sous le nom canonique /**metrics**. Lorsque vous demandez des métriques, Prometheus récupère directement les métriques à partir du point de terminaison des métriques exposé par les machines virtuelles et présente ces métriques pour affichage.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur disposant des privilèges **cluster-admin** ou du rôle **monitoring-edit**.
- Vous avez activé la surveillance du projet défini par l'utilisateur en configurant le service node-exporter.

Procédure

1. Obtenir le point de terminaison du service HTTP en spécifiant l'espace de noms du service :

```
$ oc get service -n <namespace> <node-exporter-service>
```

2. Pour obtenir la liste de toutes les mesures disponibles pour le service node-exporter, interrogez la ressource **metrics**.

```
$ curl http://<172.30.226.162:9100>/metrics | grep -vE "^#|^$"
```

Exemple de sortie

```
node_arp_entries{device="eth0"} 1
node_boot_time_seconds 1.643153218e+09
node_context_switches_total 4.4938158e+07
node_cooling_device_cur_state{name="0",type="Processor"} 0
node_cooling_device_max_state{name="0",type="Processor"} 0
node_cpu_guest_seconds_total{cpu="0",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="0",mode="user"} 0
node_cpu_seconds_total{cpu="0",mode="idle"} 1.10586485e+06
node_cpu_seconds_total{cpu="0",mode="iowait"} 37.61
node_cpu_seconds_total{cpu="0",mode="irq"} 233.91
node_cpu_seconds_total{cpu="0",mode="nice"} 551.47
node_cpu_seconds_total{cpu="0",mode="softirq"} 87.3
node_cpu_seconds_total{cpu="0",mode="steal"} 86.12
node_cpu_seconds_total{cpu="0",mode="system"} 464.15
node_cpu_seconds_total{cpu="0",mode="user"} 1075.2
node_disk_discard_time_seconds_total{device="vda"} 0
node_disk_discard_time_seconds_total{device="vdb"} 0
node_disk_discarded_sectors_total{device="vda"} 0
node_disk_discarded_sectors_total{device="vdb"} 0
node_disk_discards_completed_total{device="vda"} 0
```

```
node_disk_discards_completed_total{device="vdb"} 0
node_disk_discards_merged_total{device="vda"} 0
node_disk_discards_merged_total{device="vdb"} 0
node_disk_info{device="vda",major="252",minor="0"} 1
node_disk_info{device="vdb",major="252",minor="16"} 1
node_disk_io_now{device="vda"} 0
node_disk_io_now{device="vdb"} 0
node_disk_io_time_seconds_total{device="vda"} 174
node_disk_io_time_seconds_total{device="vdb"} 0.054
node_disk_io_time_weighted_seconds_total{device="vda"} 259.79200000000003
node_disk_io_time_weighted_seconds_total{device="vdb"} 0.039
node_disk_read_bytes_total{device="vda"} 3.71867136e+08
node_disk_read_bytes_total{device="vdb"} 366592
node_disk_read_time_seconds_total{device="vda"} 19.128
node_disk_read_time_seconds_total{device="vdb"} 0.039
node_disk_reads_completed_total{device="vda"} 5619
node_disk_reads_completed_total{device="vdb"} 96
node_disk_reads_merged_total{device="vda"} 5
node_disk_reads_merged_total{device="vdb"} 0
node_disk_write_time_seconds_total{device="vda"} 240.66400000000002
node_disk_write_time_seconds_total{device="vdb"} 0
node_disk_writes_completed_total{device="vda"} 71584
node_disk_writes_completed_total{device="vdb"} 0
node_disk_writes_merged_total{device="vda"} 19761
node_disk_writes_merged_total{device="vdb"} 0
node_disk_written_bytes_total{device="vda"} 2.007924224e+09
node_disk_written_bytes_total{device="vdb"} 0
```

14.13.4. Création d'une ressource ServiceMonitor pour le service d'exportation de nœuds

Vous pouvez utiliser une bibliothèque client Prometheus et récupérer des métriques à partir du point de terminaison /**metrics** pour accéder aux métriques exposées par le service node-exporter et les afficher. Utilisez une définition de ressource personnalisée (CRD) **ServiceMonitor** pour surveiller le service d'exportation de nœuds.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur disposant des privilèges **cluster-admin** ou du rôle **monitoring-edit**.
- Vous avez activé la surveillance du projet défini par l'utilisateur en configurant le service node-exporter.

Procédure

1. Créez un fichier YAML pour la configuration de la ressource **ServiceMonitor**. Dans cet exemple, le moniteur de services correspond à tout service portant l'étiquette **metrics** et interroge le port **exmet** toutes les 30 secondes.

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    k8s-app: node-exporter-metrics-monitor
```

```
  name: node-exporter-metrics-monitor (1)
  namespace: dynamation (2)
spec:
  endpoints:
  - interval: 30s (3)
    port: exmet (4)
    scheme: http
  selector:
    matchLabels:
      servicetype: metrics
```

1. Le nom du site **ServiceMonitor**.
2. L'espace de noms dans lequel le site **ServiceMonitor** est créé.
3. Intervalle auquel le port sera interrogé.
4. Le nom du port qui est interrogé toutes les 30 secondes

2. Créez la configuration **ServiceMonitor** pour le service node-exporter.

```
$ oc create -f node-exporter-metrics-monitor.yaml
```

14.13.4.1. Accès au service d'exportateur de nœuds en dehors du cluster

Vous pouvez accéder au service node-exporter en dehors du cluster et visualiser les métriques exposées.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur disposant des privilèges **cluster-admin** ou du rôle **monitoring-edit**.
- Vous avez activé la surveillance du projet défini par l'utilisateur en configurant le service node-exporter.

Procédure

1. Exposer le service node-exporter.

```
$ oc expose service -n <namespace> <node_exporter_service_name>
```

2. Obtenir le FQDN (Fully Qualified Domain Name) pour l'itinéraire.

```
$ oc get route -o=custom-columns=NAME:.metadata.name,DNS:.spec.host
```

Exemple de sortie

```
NAME                 DNS
node-exporter-service   node-exporter-service-dynamation.apps.cluster.example.org
```

3. Utilisez la commande **curl** pour afficher les mesures du service node-exporter.

```
$ curl -s http://node-exporter-service-dynamation.apps.cluster.example.org/metrics
```

Exemple de sortie

```
go_gc_duration_seconds{quantile="0"} 1.5382e-05
go_gc_duration_seconds{quantile="0.25"} 3.1163e-05
go_gc_duration_seconds{quantile="0.5"} 3.8546e-05
go_gc_duration_seconds{quantile="0.75"} 4.9139e-05
go_gc_duration_seconds{quantile="1"} 0.000189423
```

14.13.5. Ressources supplémentaires

- Configuration de la pile de surveillance
- Permettre le suivi de projets définis par l'utilisateur
- Gestion des indicateurs
- Examen des tableaux de bord de suivi
- Contrôler l'état de santé des applications à l'aide de bilans de santé
- Créer et utiliser des cartes de configuration
- Contrôler les états des machines virtuelles

14.14. RUNBOOKS DE VIRTUALISATION OPENSHIFT

Vous pouvez utiliser les procédures de ces runbooks pour diagnostiquer et résoudre les problèmes qui déclenchent des alertes OpenShift Virtualization.

Les alertes OpenShift Virtualization sont affichées sur la page **Virtualization** > **Overview**.

14.14.1. CDIDataImportCronOutdated

Signification
Cette alerte se déclenche lorsque **DataImportCron** ne peut pas interroger ou importer les dernières versions des images de disque.

DataImportCron ce processus permet de mettre à jour les images de disques Polls, de vérifier les dernières versions et d'importer les images en tant que réclamations de volumes persistants (PVC). Ce processus garantit que les PVC sont mis à jour à la dernière version afin qu'ils puissent être utilisés comme sources de clone fiables ou comme images dorées pour les machines virtuelles (VM).

Pour les images dorées, *latest* fait référence au dernier système d'exploitation de la distribution. Pour les autres images de disque, *latest* renvoie au dernier hachage de l'image disponible.

Impact
Les machines virtuelles peuvent être créées à partir d'images de disque obsolètes.

Les machines virtuelles peuvent ne pas démarrer parce qu'aucun PVC source n'est disponible pour le clonage.

Diagnostic

1. Vérifiez que le cluster dispose d'une classe de stockage par défaut :

```
$ oc get sc
```

   La sortie affiche les classes de stockage avec **(default)** à côté du nom de la classe de stockage par défaut. Vous devez définir une classe de stockage par défaut, soit sur le cluster, soit dans la spécification **DataImportCron**, pour que **DataImportCron** interroge et importe des images dorées. Si aucune classe de stockage n'est définie, le contrôleur DataVolume ne parvient pas à créer de PVC et l'événement suivant s'affiche : **DataVolume.storage spec is missing accessMode and no storageClass to choose profile**.

2. Obtenir l'espace de noms et le nom de **DataImportCron**:

```
$ oc get dataimportcron -A -o json | jq -r '.items[] | \
  select(.status.conditions[] | select(.type == "UpToDate" and \
  .status == "False")) | .metadata.namespace + "/" + .metadata.name'
```

3. Si une classe de stockage par défaut n'est pas définie sur le cluster, vérifiez la spécification **DataImportCron** pour une classe de stockage par défaut :

```
$ oc get dataimportcron <dataimportcron> -o yaml | \
  grep -B 5 storageClassName
```

   Exemple de sortie

```
      url: docker://.../cdi-func-test-tinycore
    storage:
      resources:
        requests:
          storage: 5Gi
    storageClassName: rook-ceph-block
```

4. Obtenir le nom de l'objet **DataVolume** associé à l'objet **DataImportCron**:

```
$ oc -n <namespace> get dataimportcron <dataimportcron> -o json | \
  jq .status.lastImportedPVC.name
```

5. Vérifiez les messages d'erreur dans le journal **DataVolume**:

```
oc -n <namespace> get dv <datavolume> -o yaml
```

6. Définir la variable d'environnement **CDI_NAMESPACE**:

```
$ export CDI_NAMESPACE="$(oc get deployment -A | \
  grep cdi-operator | awk '{print $1}')"
```

7. Vérifiez les messages d'erreur dans le journal **cdi-deployment**:

```
$ oc logs -n $CDI_NAMESPACE deployment/cdi-deployment
```

Atténuation

1. Définissez une classe de stockage par défaut, soit sur le cluster, soit dans la spécification **DataImportCron**, pour interroger et importer les images dorées. La mise à jour de Containerized Data Importer (CDI) résoudra le problème en quelques secondes.

2. Si le problème n'est pas résolu, supprimez les volumes de données associés aux objets **DataImportCron** concernés. Le CDI recréera les volumes de données avec la classe de stockage par défaut.

3. Si votre cluster est installé dans un environnement réseau restreint, désactivez la fonctionnalité **enableCommonBootImageImport** afin de ne pas recevoir de mises à jour automatiques :

```
$ oc patch hco kubevirt-hyperconverged -n $CDI_NAMESPACE --type json \
  -p '[{"op": "replace", "path": \
  "/spec/featureGates/enableCommonBootImageImport", "value": false}]'
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.2. CDIDataVolumeUnusualRestartCount

Signification
Cette alerte se déclenche lorsqu'un objet **DataVolume** redémarre plus de trois fois.

Impact
Les volumes de données sont responsables de l'importation et de la création d'un disque de machine virtuelle sur une revendication de volume persistant. Si un volume de données redémarre plus de trois fois, il est peu probable que ces opérations aboutissent. Vous devez diagnostiquer et résoudre le problème.

Diagnostic

1. Obtenir le nom et l'espace de noms du volume de données :

```
$ oc get dv -A -o json | jq -r '.items[] | \
  select(.status.restartCount>3)' | jq '.metadata.name, .metadata.namespace'
```

2. Vérifiez l'état des pods associés au volume de données :

```
$ oc get pods -n <namespace> -o json | jq -r '.items[] | \
  select(.metadata.ownerReferences[] | \
  select(.name=="<dv_name>")).metadata.name'
```

3. Obtenir les coordonnées des nacelles :

```
oc -n <namespace> describe pods <pod>
```

4. Vérifiez les messages d'erreur dans les journaux de pods :

```
oc -n <namespace> describe logs <pod>
```

Atténuation
Supprimez le volume de données, résolvez le problème et créez un nouveau volume de données.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.3. CDINotReady

Signification
Cette alerte se déclenche lorsque l'importateur de données conteneurisées (CDI) est dans un état dégradé :

- Pas de progrès
- Non disponible à l'utilisation

Impact
CDI n'est pas utilisable, les utilisateurs ne peuvent donc pas construire de disques de machines virtuelles sur des réclamations de volumes persistants (PVC) en utilisant les volumes de données de CDI. Les composants de CDI ne sont pas prêts et ils ont cessé de progresser vers un état prêt.

Diagnostic

1. Définir la variable d'environnement **CDI_NAMESPACE**:

```
$ export CDI_NAMESPACE="$(oc get deployment -A | \
  grep cdi-operator | awk '{print $1}')"
```

2. Vérifier le déploiement du CDI pour les composants qui ne sont pas prêts :

```
$ oc -n $CDI_NAMESPACE get deploy -l cdi.kubevirt.io
```

3. Vérifier les détails du pod défaillant :

```
oc -n $CDI_NAMESPACE describe pods <pod>
```

4. Vérifiez les journaux du module défaillant :

```
$ oc -n $CDI_NAMESPACE logs <pod>
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.4. CDIOperatorDown

Signification
Cette alerte se déclenche lorsque l'opérateur de l'importateur de données conteneurisées (CDI) est hors service. L'opérateur CDI déploie et gère les composants de l'infrastructure CDI, tels que les contrôleurs de volumes de données et de réclamations de volumes persistants (PVC). Ces contrôleurs aident les utilisateurs à construire des disques de machines virtuelles sur des PVC.

Impact
Les composants CDI peuvent ne pas se déployer ou rester dans un état requis. L'installation du CDI peut ne pas fonctionner correctement.

Diagnostic

1. Définir la variable d'environnement **CDI_NAMESPACE**:

```
$ export CDI_NAMESPACE="$(oc get deployment -A | grep cdi-operator | \
  awk '{print $1}')"
```

2. Vérifier si le pod **cdi-operator** est en cours d'exécution :

```
$ oc -n $CDI_NAMESPACE get pods -l name=cdi-operator
```

3. Obtenir les coordonnées de la nacelle **cdi-operator**:

```
$ oc -n $CDI_NAMESPACE describe pods -l name=cdi-operator
```

4. Vérifiez si le journal du pod **cdi-operator** contient des erreurs :

```
$ oc -n $CDI_NAMESPACE logs -l name=cdi-operator
```

Atténuation
Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.5. CDIStorageProfilesIncomplete

Signification
Cette alerte se déclenche lorsqu'un profil de stockage Containerized Data Importer (CDI) est incomplet.

Si un profil de stockage est incomplet, le CDI ne peut pas déduire les champs PVC (persistent volume claim), tels que **volumeMode** et **accessModes**, qui sont nécessaires pour créer un disque de machine virtuelle (VM).

Impact
Le CDI ne peut pas créer de disque VM sur le PVC.

Diagnostic

- Identifier le profil de stockage incomplet :

```
oc get storageprofile <storage_class> $ oc get storageprofile <storage_class>
```

Atténuation

- Ajoutez les informations manquantes du profil de stockage comme dans l'exemple suivant :

```
$ oc patch storageprofile local --type=merge -p '{"spec": \
  {"claimPropertySets": [{"accessModes": ["ReadWriteOnce"], \
  "volumeMode": "Filesystem"}]}}'
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.6. CnaoDown

Signification
Cette alerte se déclenche lorsque le Cluster Network Addons Operator (CNAO) est hors service. Le CNAO déploie des composants réseau supplémentaires au-dessus du cluster.

Impact
Si l'ACAO n'est pas en cours d'exécution, le cluster ne peut pas réconcilier les modifications apportées aux composants des machines virtuelles. Par conséquent, les modifications risquent de ne pas être prises en compte.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | \
  grep cluster-network-addons-operator | awk '{print $1}')"
```

2. Vérifier l'état du pod **cluster-network-addons-operator**:

```
$ oc -n $NAMESPACE get pods -l name=cluster-network-addons-operator
```

3. Vérifiez les messages d'erreur dans les journaux de **cluster-network-addons-operator**:

```
$ oc -n $NAMESPACE logs -l name=cluster-network-addons-operator
```

4. Obtenir les informations sur les nacelles du site **cluster-network-addons-operator**:

```
$ oc -n $NAMESPACE describe pods -l name=cluster-network-addons-operator
```

Atténuation
Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.7. HPPNotReady

Signification
Cette alerte se déclenche lorsqu'une installation de Hostpath Provisioner (HPP) est dans un état dégradé.

Le HPP provisionne dynamiquement les volumes du chemin d'accès de l'hôte afin de fournir un stockage pour les réclamations de volumes persistants (PVC).

Impact
Le HPP n'est pas utilisable. Ses composants ne sont pas prêts et ne progressent pas vers un état de préparation.

Diagnostic

1. Définir la variable d'environnement **HPP_NAMESPACE**:

```
$ export HPP_NAMESPACE="$(oc get deployment -A | \
  grep hostpath-provisioner-operator | awk '{print $1}')"
```

2. Vérifier les composants HPP qui ne sont pas encore prêts :

```
$ oc -n $HPP_NAMESPACE get all -l k8s-app=hostpath-provisioner
```

3. Obtenir les coordonnées de la nacelle défaillante :

```
oc -n $HPP_NAMESPACE describe pods <pod>
```

4. Vérifiez les journaux du module défaillant :

```
$ oc -n $HPP_NAMESPACE logs <pod>
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.8. HPPOperatorDown

Signification
Cette alerte est déclenchée lorsque l'opérateur HPP (hostpath provisioner) est en panne.

L'opérateur HPP déploie et gère les composants de l'infrastructure HPP, tels que le jeu de démons qui approvisionne les volumes des chemins d'accès à l'hôte.

Impact
Les composants HPP risquent de ne pas se déployer ou de ne pas rester dans l'état requis. Par conséquent, l'installation HPP peut ne pas fonctionner correctement dans le cluster.

Diagnostic

1. Configurez la variable d'environnement **HPP_NAMESPACE**:

```
$ HPP_NAMESPACE="$(oc get deployment -A | grep \
  hostpath-provisioner-operator | awk '{print $1}')"
```

2. Vérifier si le pod **hostpath-provisioner-operator** est en cours d'exécution :

```
$ oc -n $HPP_NAMESPACE get pods -l name=hostpath-provisioner-operator
```

3. Obtenir les coordonnées de la nacelle **hostpath-provisioner-operator**:

```
$ oc -n $HPP_NAMESPACE describe pods -l name=hostpath-provisioner-operator
```

4. Vérifiez si le journal du pod **hostpath-provisioner-operator** contient des erreurs :

```
$ oc -n $HPP_NAMESPACE logs -l name=hostpath-provisioner-operator
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.9. HPPSharingPoolPathWithOS

Signification
Cette alerte se déclenche lorsque le hostpath provisioner (HPP) partage un système de fichiers avec d'autres composants critiques, tels que **kubelet** ou le système d'exploitation (OS).

HPP fournit dynamiquement des volumes de chemin d'accès à l'hôte pour assurer le stockage des réclamations de volumes persistants (PVC).

Impact
Un pool de chemins d'accès partagés exerce une pression sur les disques du nœud. Les performances et la stabilité du nœud peuvent se dégrader.

Diagnostic

1. Configurez la variable d'environnement **HPP_NAMESPACE**:

```
$ export HPP_NAMESPACE="$(oc get deployment -A | \
  grep hostpath-provisioner-operator | awk '{print $1}')"
```

2. Obtenir l'état du démon **hostpath-provisioner-csi** set pods :

```
$ oc -n $HPP_NAMESPACE get pods | grep hostpath-provisioner-csi
```

3. Consultez les journaux de **hostpath-provisioner-csi** pour identifier le pool partagé et le chemin d'accès :

```
oc -n $HPP_NAMESPACE logs <csi_daemonset> -c hostpath-provisioner
```

Exemple de sortie

```
I0208 15:21:03.769731       1 utils.go:221] pool (<legacy, csi-data-dir>/csi),
shares path with OS which can lead to node disk pressure
```

Atténuation
À l'aide des données obtenues dans la section Diagnostic, essayez d'empêcher le partage du chemin d'accès au pool avec le système d'exploitation. Les étapes spécifiques varient en fonction du nœud et d'autres circonstances.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.10. KubeMacPoolDown

Signification
KubeMacPool est en panne. **KubeMacPool** est responsable de l'attribution des adresses MAC et de la prévention des conflits d'adresses MAC.

Impact
Si **KubeMacPool** est en panne, les objets **VirtualMachine** ne peuvent pas être créés.

Diagnostic

1. Définir la variable d'environnement **KMP_NAMESPACE**:

```
$ export KMP_NAMESPACE="$(oc get pod -A --no-headers -l \
  control-plane=mac-controller-manager | awk '{print $1}')"
```

2. Définir la variable d'environnement **KMP_NAME**:

```
$ export KMP_NAME="$(oc get pod -A --no-headers -l \
  control-plane=mac-controller-manager | awk '{print $2}')"
```

3. Obtenir les coordonnées du pod **KubeMacPool-manager**:

```
$ oc describe pod -n $KMP_NAMESPACE $KMP_NAME
```

4. Vérifiez les messages d'erreur dans les journaux de **KubeMacPool-manager**:

```
$ oc logs -n $KMP_NAMESPACE $KMP_NAME
```

Atténuation
Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.11. KubeMacPoolDuplicateMacsFound

Signification
Cette alerte se déclenche lorsque **KubeMacPool** détecte des adresses MAC en double.

KubeMacPool est responsable de l'attribution des adresses MAC et de la prévention des conflits d'adresses MAC. Au démarrage de **KubeMacPool**, il recherche dans le cluster les adresses MAC des machines virtuelles (VM) dans les espaces de noms gérés.

Impact
Les adresses MAC en double sur le même réseau local peuvent causer des problèmes de réseau.

Diagnostic

1. Obtenir l'espace de noms et le nom du pod **kubemacpool-mac-controller**:

```
$ oc get pod -A -l control-plane=mac-controller-manager --no-headers \
  -o custom-columns=":metadata.namespace,:metadata.name"
```

2. Obtenir les adresses MAC dupliquées à partir des journaux **kubemacpool-mac-controller**:

```
$ oc logs -n <namespace> <kubemacpool_mac_controller> | \
  grep "already allocated"
```

Exemple de sortie

```
mac address 02:00:ff:ff:ff:ff already allocated to
vm/kubemacpool-test/testvm, br1,
conflict with: vm/kubemacpool-test/testvm2, br1
```

Atténuation

1. Mettez à jour les machines virtuelles pour supprimer les adresses MAC en double.
2. Redémarrez le pod **kubemacpool-mac-controller**:

```
oc delete pod -n <namespace> <kubemacpool_mac_controller>
```

14.14.12. KubeVirtComponentExceedsRequestedCPU (composant KubeVirt)

Signification
Cette alerte se déclenche lorsque l'utilisation de l'unité centrale d'un composant dépasse la limite demandée.

Impact
L'utilisation des ressources de l'unité centrale n'est pas optimale et le nœud pourrait être surchargé.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier la limite de demande de l'unité centrale du composant :

```
oc -n $NAMESPACE get deployment <component> -o yaml | grep requests : -A 2
```

3. Vérifiez l'utilisation réelle de l'unité centrale à l'aide d'une requête PromQL :

```
node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate
{namespace="$NAMESPACE",container="<component>"}
```

Voir la documentation Prometheus pour plus d'informations.

Atténuation
Mettre à jour la limite de demande de CPU dans la ressource personnalisée **HCO**.

14.14.13. KubeVirtComponentExceedsRequestedMemory (Dépasse la mémoire demandée)

Signification
Cette alerte se déclenche lorsque l'utilisation de la mémoire d'un composant dépasse la limite demandée.

Impact
L'utilisation des ressources mémoire n'est pas optimale et le nœud peut être surchargé.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier la limite de demande de mémoire du composant :

```
$ oc -n $NAMESPACE get deployment <component> -o yaml | \
  grep requests: -A 2
```

3. Vérifiez l'utilisation réelle de la mémoire à l'aide d'une requête PromQL :

```
container_memory_usage_bytes{namespace="$NAMESPACE",container="<component>"}
```

Voir la documentation Prometheus pour plus d'informations.

Atténuation
Mettre à jour la limite de demande de mémoire dans la ressource personnalisée **HCO**.

14.14.14. KubevirtHyperconvergedClusterOperatorCRModification

Signification
Cette alerte se déclenche lorsqu'un opérande de l'opérateur de cluster hyperconvergé (HCO) est modifié par quelqu'un ou quelque chose d'autre que HCO.

HCO configure OpenShift Virtualization et ses opérateurs de support d'une manière fondée sur l'opinion et écrase ses opérandes lorsqu'il y a un changement inattendu à leur égard. Les utilisateurs ne doivent pas modifier les opérandes directement. La ressource personnalisée **HyperConverged** est la source de vérité pour la configuration.

Impact
La modification manuelle des opérandes entraîne une fluctuation de la configuration du cluster et peut conduire à une instabilité.

Diagnostic

- Vérifiez la valeur **component_name** dans les détails de l'alerte pour déterminer le type d'opérande (**kubevirt**) et le nom de l'opérande (**kubevirt-kubevirt-hyperconverged**) qui sont modifiés :

```
Labels
  alertname=KubevirtHyperconvergedClusterOperatorCRModification
  component_name=kubevirt/kubevirt-kubevirt-hyperconverged
  severity=warning
```

Atténuation
Ne modifiez pas directement les opérandes HCO. Utilisez les objets **HyperConverged** pour configurer le cluster.

L'alerte se résout d'elle-même après 10 minutes si les opérandes ne sont pas modifiés manuellement.

14.14.15. KubevirtHyperconvergedClusterOperatorInstallationNotCompletedAlert (alerte d'installation non terminée)

Signification
Cette alerte se déclenche lorsque l'opérateur de cluster hyperconvergé (HCO) fonctionne pendant plus d'une heure sans ressource personnalisée (CR) **HyperConverged**.

Cette alerte a les causes suivantes :

- Au cours de la procédure d'installation, vous avez installé le HCO mais vous n'avez pas créé le CR **HyperConverged**.
- Au cours du processus de désinstallation, vous avez supprimé le CR **HyperConverged** avant de désinstaller le HCO et le HCO est toujours en cours d'exécution.

Atténuation
L'atténuation dépend de l'installation ou de la désinstallation du HCO :

- Terminez l'installation en créant un CR **HyperConverged** avec ses valeurs par défaut :

```
$ cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: hco-operatorgroup
  namespace: kubevirt-hyperconverged
spec: {}
EOF
```

- Désinstallez le HCO. Si le processus de désinstallation continue de s'exécuter, vous devez résoudre ce problème afin d'annuler l'alerte.

14.14.16. KubevirtHyperconvergedClusterOperatorUSModification

Signification
Cette alerte se déclenche lorsqu'une annotation JSON Patch est utilisée pour modifier un opérande de l'opérateur de cluster hyperconvergé (HCO).

HCO configure OpenShift Virtualization et ses opérateurs de support d'une manière fondée sur l'opinion et écrase ses opérandes lorsqu'il y a un changement inattendu à leur égard. Les utilisateurs ne doivent pas modifier les opérandes directement.

Toutefois, si une modification est nécessaire et qu'elle n'est pas prise en charge par l'API du HCO, vous pouvez forcer le HCO à définir une modification dans un opérateur à l'aide d'annotations JSON Patch. Ces modifications ne sont pas annulées par le HCO au cours de son processus de réconciliation.

Impact
L'utilisation incorrecte des annotations JSON Patch peut conduire à des résultats inattendus ou à un environnement instable.

La mise à jour d'un système avec des annotations JSON Patch est dangereuse car la structure des ressources personnalisées des composants peut changer.

Diagnostic

- Consultez le site **annotation_name** dans les détails de l'alerte pour identifier l'annotation JSON Patch :

```
Labels
  alertname=KubevirtHyperconvergedClusterOperatorUSModification
  annotation_name=kubevirt.kubevirt.io/jsonpatch
  severity=info
```

Atténuation

Il est préférable d'utiliser l'API HCO pour modifier un opérande. Toutefois, si la modification ne peut être effectuée qu'à l'aide d'une annotation JSON Patch, il convient de procéder avec prudence.

Supprimer les annotations JSON Patch avant la mise à jour pour éviter les problèmes potentiels.

14.14.17. KubevirtVmHighMemoryUsage

Signification
Cette alerte se déclenche lorsqu'un conteneur hébergeant une machine virtuelle (VM) dispose de moins de 20 Mo de mémoire libre.

Impact
La machine virtuelle s'exécutant dans le conteneur est arrêtée par le moteur d'exécution si la limite de mémoire du conteneur est dépassée.

Diagnostic

1. Obtenir les coordonnées du pod **virt-launcher**:

```
$ oc get pod <virt-launcher> -o yaml
```

2. Identifier les processus du conteneur **compute** qui utilisent beaucoup de mémoire dans le pod **virt-launcher**:

```
oc exec -it <virt-launcher> -c compute -- top
```

Atténuation

- Augmentez la limite de mémoire dans la spécification **VirtualMachine** comme dans l'exemple suivant :

```
spec:
  running: false
  template:
    metadata:
      labels:
        kubevirt.io/vm: vm-name
    spec:
      domain:
        resources:
          limits:
            memory: 200Mi
          requests:
            memory: 128Mi
```

14.14.18. KubeVirtVMIExcessiveMigrations

Signification
Cette alerte se déclenche lorsqu'une instance de machine virtuelle (VMI) migre en direct plus de 12 fois sur une période de 24 heures.

Ce taux de migration est anormalement élevé, même lors d'une mise à niveau. Cette alerte peut indiquer un problème dans l'infrastructure du cluster, comme des perturbations du réseau ou des ressources insuffisantes.

Impact
Une machine virtuelle (VM) qui migre trop fréquemment risque de voir ses performances se dégrader car des erreurs de page mémoire se produisent pendant la transition.

Diagnostic

1. Vérifiez que le nœud de travail dispose de ressources suffisantes :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | \
  jq .items[].status.allocatable
```

Exemple de sortie

```
{
  "cpu": "3500m",
  "devices.kubevirt.io/kvm": "1k",
  "devices.kubevirt.io/sev": "0",
  "devices.kubevirt.io/tun": "1k",
  "devices.kubevirt.io/vhost-net": "1k",
  "ephemeral-storage": "38161122446",
  "hugepages-1Gi": "0",
  "hugepages-2Mi": "0",
  "memory": "7000128Ki",
  "pods": "250"
}
```

2. Vérifier l'état du nœud de travail :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | \
  jq .items[].status.conditions
```

Exemple de sortie

```
{
  "lastHeartbeatTime": "2022-05-26T07:36:01Z",
  "lastTransitionTime": "2022-05-23T08:12:02Z",
  "message": "kubelet has sufficient memory available",
  "reason": "KubeletHasSufficientMemory",
  "status": "False",
  "type": "MemoryPressure"
},
{
  "lastHeartbeatTime": "2022-05-26T07:36:01Z",
  "lastTransitionTime": "2022-05-23T08:12:02Z",
  "message": "kubelet has no disk pressure",
  "reason": "KubeletHasNoDiskPressure",
  "status": "False",
  "type": "DiskPressure"
},
{
  "lastHeartbeatTime": "2022-05-26T07:36:01Z",
  "lastTransitionTime": "2022-05-23T08:12:02Z",
  "message": "kubelet has sufficient PID available",
  "reason": "KubeletHasSufficientPID",
  "status": "False",
```

```
    "type": "PIDPressure"
  },
  {
    "lastHeartbeatTime": "2022-05-26T07:36:01Z",
    "lastTransitionTime": "2022-05-23T08:24:15Z",
    "message": "kubelet is posting ready status",
    "reason": "KubeletReady",
    "status": "True",
    "type": "Ready"
  }
```


3. Connectez-vous au nœud de travail et vérifiez que le service **kubelet** est en cours d'exécution :

```
$ systemctl status kubelet
```

4. Vérifiez les messages d'erreur dans le journal **kubelet**:

```
$ journalctl -r -u kubelet
```

Atténuation
Assurez-vous que les nœuds de travail disposent de ressources suffisantes (CPU, mémoire, disque) pour exécuter les charges de travail des VM sans interruption.

Si le problème persiste, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.19. KubeVirtVMStuckInErrorState

Signification
Cette alerte se déclenche lorsqu'une machine virtuelle (VM) est dans un état d'erreur pendant plus de 5 minutes.

États d'erreur :

- CrashLoopBackOff
- Inconnu
- Insaisissable
- ErrImagePull
- ImagePullBackOff
- PvcNotFound
- Erreur de volume de données

Cette alerte peut indiquer un problème au niveau de la configuration de la VM, par exemple une réclamation de volume persistant manquante, ou un problème au niveau de l'infrastructure du cluster, par exemple des perturbations du réseau ou des ressources insuffisantes au niveau des nœuds.

Impact

Il n'y a pas d'impact immédiat. Cependant, si l'alerte persiste, vous devez rechercher la cause première et résoudre le problème.

Diagnostic

1. Vérifiez les détails de l'instance de machine virtuelle (VMI) :

```
$ oc describe vmi <vmi> -n <namespace>
```

Exemple de sortie

```
Name:         testvmi-hxghp
Namespace:    kubevirt-test-default1
Labels:       name=testvmi-hxghp
Annotations:  kubevirt.io/latest-observed-api-version: v1
              kubevirt.io/storage-observed-api-version: v1alpha3
API Version:  kubevirt.io/v1
Kind:         VirtualMachineInstance
...
Spec:
  Domain:
...
    Resources:
      Requests:
        Cpu:     5000000Gi
        Memory:  5130000240Mi
...
Status:
...
  Conditions:
    Last Probe Time:       2022-10-03T11:11:07Z
    Last Transition Time:  2022-10-03T11:11:07Z
    Message:               Guest VM is not reported as running
    Reason:                GuestNotRunning
    Status:                False
    Type:                  Ready
    Last Probe Time:       <nil>
    Last Transition Time:  2022-10-03T11:11:07Z
    Message:               0/2 nodes are available: 2 Insufficient cpu, 2
      Insufficient memory.
    Reason:                Unschedulable
    Status:                False
    Type:                  PodScheduled
  Guest OS Info:
  Phase:  Scheduling
  Phase Transition Timestamps:
    Phase:                     Pending
    Phase Transition Timestamp:  2022-10-03T11:11:07Z
    Phase:                     Scheduling
    Phase Transition Timestamp:  2022-10-03T11:11:07Z
  Qos Class:                   Burstable
  Runtime User:                0
  Virtual Machine Revision Name:  revision-start-vm-3503e2dc-27c0-46ef-9167-
7ae2e7d93e6e-1
Events:
  Type    Reason          Age   From                    Message
```

```
----    ------           ----  ----                    -------
Normal  SuccessfulCreate  27s   virtualmachine-controller  Created virtual
  machine pod virt-launcher-testvmi-hxghp-xh9qn
```

2. Vérifier les ressources du nœud :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | jq '.items | \
  .[].status.allocatable'
```

Exemple de sortie

```
{
  "cpu": "5",
  "devices.kubevirt.io/kvm": "1k",
  "devices.kubevirt.io/sev": "0",
  "devices.kubevirt.io/tun": "1k",
  "devices.kubevirt.io/vhost-net": "1k",
  "ephemeral-storage": "33812468066",
  "hugepages-1Gi": "0",
  "hugepages-2Mi": "128Mi",
  "memory": "3783496Ki",
  "pods": "110"
}
```

3. Vérifier que le nœud ne présente pas de conditions d'erreur :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | jq '.items | \
  .[].status.conditions'
```

Exemple de sortie

```
[
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has sufficient memory available",
    "reason": "KubeletHasSufficientMemory",
    "status": "False",
    "type": "MemoryPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has no disk pressure",
    "reason": "KubeletHasNoDiskPressure",
    "status": "False",
    "type": "DiskPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has sufficient PID available",
    "reason": "KubeletHasSufficientPID",
    "status": "False",
```

```
    "type": "PIDPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:30Z",
    "message": "kubelet is posting ready status",
    "reason": "KubeletReady",
    "status": "True",
    "type": "Ready"
  }
]
```


Atténuation
Essayez d'identifier et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.20. KubeVirtVMStuckInMigratingState

Signification
Cette alerte se déclenche lorsqu'une machine virtuelle (VM) est en état de migration pendant plus de 5 minutes.

Cette alerte peut indiquer un problème dans l'infrastructure de la grappe, comme des perturbations du réseau ou des ressources insuffisantes au niveau des nœuds.

Impact
Il n'y a pas d'impact immédiat. Cependant, si l'alerte persiste, vous devez rechercher la cause première et résoudre le problème.

Diagnostic

1. Vérifier les ressources du nœud :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | jq '.items | \
  .[].status.allocatable'
```

Exemple de sortie


```
{
  "cpu": "5",
  "devices.kubevirt.io/kvm": "1k",
  "devices.kubevirt.io/sev": "0",
  "devices.kubevirt.io/tun": "1k",
  "devices.kubevirt.io/vhost-net": "1k",
  "ephemeral-storage": "33812468066",
  "hugepages-1Gi": "0",
  "hugepages-2Mi": "128Mi",
  "memory": "3783496Ki",
  "pods": "110"
}
```


2. Vérifier les conditions d'état du nœud :

```
$ oc get nodes -l node-role.kubernetes.io/worker= -o json | jq '.items | \
  .[].status.conditions'
```

Exemple de sortie


```
[
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has sufficient memory available",
    "reason": "KubeletHasSufficientMemory",
    "status": "False",
    "type": "MemoryPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has no disk pressure",
    "reason": "KubeletHasNoDiskPressure",
    "status": "False",
    "type": "DiskPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:20Z",
    "message": "kubelet has sufficient PID available",
    "reason": "KubeletHasSufficientPID",
    "status": "False",
    "type": "PIDPressure"
  },
  {
    "lastHeartbeatTime": "2022-10-03T11:13:34Z",
    "lastTransitionTime": "2022-10-03T10:14:30Z",
    "message": "kubelet is posting ready status",
    "reason": "KubeletReady",
    "status": "True",
    "type": "Ready"
  }
]
```


Atténuation

Vérifiez la configuration de la migration de la machine virtuelle pour vous assurer qu'elle est adaptée à la charge de travail.

Vous définissez une configuration de migration à l'échelle du cluster en modifiant la strophe **MigrationConfiguration** de la ressource personnalisée **KubeVirt**.

Vous définissez une configuration de migration pour une portée spécifique en créant une politique de migration.

Vous pouvez déterminer si une VM est liée à une politique de migration en consultant son paramètre **vm.Status.MigrationState.MigrationPolicyName**.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.21. KubeVirtVMStuckInStartingState (état de démarrage)

Signification
Cette alerte se déclenche lorsqu'une machine virtuelle (VM) est dans un état de démarrage pendant plus de 5 minutes.

Cette alerte peut indiquer un problème dans la configuration de la VM, comme une classe de priorité mal configurée ou un périphérique réseau manquant.

Impact
Il n'y a pas d'impact immédiat. Cependant, si l'alerte persiste, vous devez rechercher la cause première et résoudre le problème.

Diagnostic

- Vérifiez les détails de l'instance de machine virtuelle (VMI) pour les conditions d'erreur :

```
$ oc describe vmi <vmi> -n <namespace>
```

Exemple de sortie

```
Name:         testvmi-ldgrw
Namespace:    kubevirt-test-default1
Labels:       name=testvmi-ldgrw
Annotations:  kubevirt.io/latest-observed-api-version: v1
              kubevirt.io/storage-observed-api-version: v1alpha3
API Version:  kubevirt.io/v1
Kind:         VirtualMachineInstance
...
Spec:
...
  Networks:
    Name:  default
    Pod:
  Priority Class Name:              non-preemtible
  Termination Grace Period Seconds:  0
Status:
  Conditions:
    Last Probe Time:       2022-10-03T11:08:30Z
    Last Transition Time:  2022-10-03T11:08:30Z
    Message:               virt-launcher pod has not yet been scheduled
    Reason:                PodNotExists
    Status:                False
    Type:                  Ready
    Last Probe Time:       <nil>
    Last Transition Time:  2022-10-03T11:08:30Z
    Message:               failed to create virtual machine pod: pods
    "virt-launcher-testvmi-ldgrw-" is forbidden: no PriorityClass with name
    non-preemtible was found
    Reason:                FailedCreate
    Status:                False
    Type:                  Synchronized
  Guest OS Info:
  Phase:  Pending
  Phase Transition Timestamps:
    Phase:                     Pending
```

```
  Phase Transition Timestamp:  2022-10-03T11:08:30Z
 Runtime User:                0
 Virtual Machine Revision Name:
  revision-start-vm-6f01a94b-3260-4c5a-bbe5-dc98d13e6bea-1
Events:
 Type     Reason        Age               From                    Message
 ----     ------        ----              ----                    -------
 Warning  FailedCreate  8s (x13 over 28s)  virtualmachine-controller  Error
 creating pod: pods "virt-launcher-testvmi-ldgrw-" is forbidden: no
 PriorityClass with name non-preemtible was found
```

Atténuation
Assurez-vous que la VM est configurée correctement et qu'elle dispose des ressources nécessaires.

L'état **Pending** indique que la VM n'a pas encore été planifiée. Vérifiez les causes possibles suivantes :

- Le pod **virt-launcher** n'est pas programmé.
- Les indications topologiques pour l'IMV ne sont pas à jour.
- Le volume de données n'est pas provisionné ou prêt.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.22. LowKVMNodesCount

Signification
Cette alerte se déclenche lorsque moins de deux nœuds de la grappe disposent de ressources KVM.

Impact
La grappe doit comporter au moins deux nœuds dotés de ressources KVM pour la migration en direct.

Les machines virtuelles ne peuvent pas être planifiées ou exécutées si aucun nœud ne dispose de ressources KVM.

Diagnostic

- Identifier les nœuds disposant de ressources KVM :

```
$ oc get nodes -o jsonpath='{.items[*].status.allocatable}' | \
  grep devices.kubevirt.io/kvm
```

Atténuation
Installez KVM sur les nœuds ne disposant pas de ressources KVM.

14.14.23. LowReadyVirtControllersCount (Nombre de contrôleurs de protection prêts à l'emploi)

Signification
Cette alerte se déclenche lorsqu'un ou plusieurs pods **virt-controller** sont en cours d'exécution, mais qu'aucun de ces pods n'a été dans l'état **Ready** au cours des 5 dernières minutes.

Un dispositif **virt-controller** surveille les définitions de ressources personnalisées (CRD) d'une instance de machine virtuelle (VMI) et gère les pods associés. Le dispositif crée des pods pour les VMI et gère

leur cycle de vie. Ce dispositif est essentiel pour la fonctionnalité de virtualisation à l'échelle de la grappe.

Impact
Cette alerte indique qu'une défaillance au niveau du cluster risque de se produire. Les actions liées à la gestion du cycle de vie des VM, telles que le lancement d'une nouvelle VMI ou l'arrêt d'une VMI existante, échoueront.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez qu'un dispositif **virt-controller** est disponible :

```
$ oc get deployment -n $NAMESPACE virt-controller \
  -o jsonpath='{.status.readyReplicas}'
```

3. Vérifiez l'état du déploiement de **virt-controller**:

```
$ oc -n $NAMESPACE get deploy virt-controller -o yaml
```

4. Obtenez les détails du déploiement de **virt-controller** pour vérifier les conditions d'état, telles que les pods en panne ou les échecs d'extraction d'images :

```
$ oc -n $NAMESPACE describe deploy virt-controller
```

5. Vérifiez si des problèmes sont survenus avec les nœuds. Par exemple, ils peuvent être dans l'état **NotReady**:

```
$ oc get nodes
```

Atténuation
Cette alerte peut avoir des causes multiples, notamment les suivantes :

- La mémoire du cluster est insuffisante.
- Les nœuds sont en panne.
- Le serveur API est surchargé. Par exemple, le planificateur peut être très sollicité et n'est donc pas entièrement disponible.
- Il y a des problèmes de réseau.

Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.24. LowReadyVirtOperatorsCount (Nombre d'opérateurs prêts à fonctionner)

Signification

Cette alerte se déclenche lorsqu'un ou plusieurs pods **virt-operator** sont en cours d'exécution, mais qu'aucun de ces pods n'a été en état **Ready** au cours des 10 dernières minutes.

Le **virt-operator** est le premier opérateur à démarrer dans un cluster. Le déploiement **virt-operator** a une réplique par défaut de deux pods **virt-operator**.

Ses principales responsabilités sont les suivantes

- Installation, mise à jour en direct et mise à niveau en direct d'un cluster
- Surveiller le cycle de vie des contrôleurs de premier niveau, tels que **virt-controller**, **virt-handler**, **virt-launcher**, et gérer leur rapprochement
- Certaines tâches à l'échelle de la grappe, telles que la rotation des certificats et la gestion de l'infrastructure

Impact
Une panne au niveau de la grappe peut se produire. Les fonctionnalités de gestion critiques à l'échelle de la grappe, telles que la rotation des certifications, la mise à niveau et le rapprochement des contrôleurs, peuvent devenir indisponibles. Un tel état déclenche également l'alerte **NoReadyVirtOperator**.

Le site **virt-operator** n'est pas directement responsable des machines virtuelles (VM) dans le cluster. Par conséquent, son indisponibilité temporaire n'affecte pas de manière significative les charges de travail des machines virtuelles.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Obtenir le nom du déploiement **virt-operator**:

```
$ oc -n $NAMESPACE get deploy virt-operator -o yaml
```

3. Obtenir les détails du déploiement de **virt-operator**:

```
$ oc -n $NAMESPACE describe deploy virt-operator
```

4. Vérifier l'absence de problèmes au niveau du nœud, tels que l'état de **NotReady**:

```
$ oc get nodes
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.25. LowVirtAPICount

Signification

Cette alerte se déclenche lorsqu'un seul pod **virt-api** disponible est détecté au cours d'une période de 60 minutes, alors qu'au moins deux nœuds sont disponibles pour la planification.

Impact
Une interruption des appels API peut se produire lors de l'éviction d'un nœud, car le pod **virt-api** devient un point de défaillance unique.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier le nombre d'unités disponibles sur le site **virt-api**:

```
$ oc get deployment -n $NAMESPACE virt-api \
  -o jsonpath='{.status.readyReplicas}'
```

3. Vérifiez l'état du déploiement de **virt-api** pour voir s'il n'y a pas de conditions d'erreur :

```
$ oc -n $NAMESPACE get deploy virt-api -o yaml
```

4. Vérifiez que les nœuds ne présentent pas de problèmes, par exemple s'ils sont dans l'état **NotReady**:

```
$ oc get nodes
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.26. LowVirtControllersCount

Signification
Cette alerte se déclenche lorsqu'un faible nombre de pods **virt-controller** est détecté. Au moins un pod **virt-controller** doit être disponible pour assurer une haute disponibilité. Le nombre de répliques par défaut est de 2.

Un dispositif **virt-controller** surveille les définitions de ressources personnalisées (CRD) d'une instance de machine virtuelle (VMI) et gère les pods associés. Le dispositif crée des pods pour les VMI et gère le cycle de vie des pods. Ce dispositif est essentiel pour la fonctionnalité de virtualisation à l'échelle du cluster.

Impact
La réactivité d'OpenShift Virtualization pourrait être affectée négativement. Par exemple, certaines demandes pourraient être manquées.

En outre, si une autre instance **virt-launcher** se termine de manière inattendue, OpenShift Virtualization peut ne plus répondre du tout.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez que les pods **virt-controller** en cours d'exécution sont disponibles :

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-controller
```

3. Vérifiez les messages d'erreur dans les journaux de **virt-launcher**:

```
oc -n $NAMESPACE logs <virt-launcher>
```

4. Obtenez les détails du pod **virt-launcher** pour vérifier les conditions d'état telles qu'une terminaison inattendue ou un état **NotReady**.

```
oc -n $NAMESPACE describe pod/<virt-launcher>
```

Atténuation
Cette alerte peut avoir diverses causes, notamment

- Mémoire insuffisante sur le cluster
- Les nœuds sont hors service
- Le serveur API est surchargé. Par exemple, le planificateur peut être très sollicité et n'est donc pas entièrement disponible.
- Questions relatives à la mise en réseau

Identifiez la cause première et corrigez-la, si possible.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.27. LowVirtOperatorCount

Signification
Cette alerte se déclenche lorsqu'un seul pod **virt-operator** dans l'état **Ready** a fonctionné au cours des 60 dernières minutes.

Le site **virt-operator** est le premier opérateur à démarrer dans une grappe. Ses principales responsabilités sont les suivantes

- Installation, mise à jour en direct et mise à niveau en direct d'un cluster
- Surveiller le cycle de vie des contrôleurs de premier niveau, tels que **virt-controller**, **virt-handler**, **virt-launcher**, et gérer leur rapprochement
- Certaines tâches à l'échelle de la grappe, telles que la rotation des certificats et la gestion de l'infrastructure

Impact
Le site **virt-operator** ne peut pas assurer la haute disponibilité (HA) du déploiement. HA nécessite au moins deux pods **virt-operator** dans un état **Ready**. Le déploiement par défaut est de deux modules.

Le site **virt-operator** n'est pas directement responsable des machines virtuelles (VM) dans le cluster. Par conséquent, sa baisse de disponibilité n'affecte pas de manière significative les charges de travail des machines virtuelles.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état des cosses **virt-operator**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-operator
```

3. Examinez les journaux des pods **virt-operator** concernés :

```
oc -n $NAMESPACE logs <virt-operator>
```

4. Obtenir les détails des pods **virt-operator** concernés :

```
oc -n $NAMESPACE describe pod <virt-operator>
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les artefacts recueillis au cours de la procédure de diagnostic.

14.14.28. NetworkAddonsConfigNotReady

Signification
Cette alerte se déclenche lorsque la ressource personnalisée (CR) **NetworkAddonsConfig** de l'opérateur de modules complémentaires du réseau de grappes (CNAO) n'est pas prête.

CNAO déploie des composants réseau supplémentaires sur le cluster. Cette alerte indique qu'un des composants déployés n'est pas prêt.

Impact
La fonctionnalité du réseau est affectée.

Diagnostic

1. Vérifiez les conditions d'état de la CR **NetworkAddonsConfig** pour identifier le déploiement ou l'ensemble de démons qui n'est pas prêt :

```
$ oc get networkaddonsconfig \
  -o custom-columns="":.status.conditions[*].message
```

Exemple de sortie

```
DaemonSet "cluster-network-addons/macvtap-cni" update is being processed...
```

2. Vérifiez que le pod du composant ne contient pas d'erreurs :

```
oc -n cluster-network-addons get daemonset <pod> -o yaml
```

3. Vérifiez les journaux du composant :

```
$ oc -n cluster-network-addons logs <pod>
```

4. Vérifier les détails du composant pour les conditions d'erreur :

```
$ oc -n cluster-network-addons describe <pod>
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.29. NoLeadingVirtOperator

Signification
Cette alerte se déclenche lorsqu'aucun pod **virt-operator** avec un bail de leader n'a été détecté pendant 10 minutes, bien que les pods **virt-operator** soient dans un état **Ready**. L'alerte indique qu'aucun pod leader n'est disponible.

Le site **virt-operator** est le premier opérateur à démarrer dans une grappe. Ses principales responsabilités sont les suivantes

- Installation, mise à jour en direct et mise à niveau en direct d'un cluster
- Surveiller le cycle de vie des contrôleurs de premier niveau, tels que **virt-controller**, **virt-handler**, **virt-launcher**, et gérer leur rapprochement
- Certaines tâches à l'échelle de la grappe, telles que la rotation des certificats et la gestion de l'infrastructure

Le déploiement **virt-operator** a une réplique par défaut de 2 pods, avec un pod détenant un bail de leader.

Impact
Cette alerte indique une défaillance au niveau de la grappe. En conséquence, les fonctionnalités critiques de gestion de la grappe, telles que la rotation des certifications, la mise à niveau et le rapprochement des contrôleurs, risquent de ne pas être disponibles.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A -o \
  custom-columns="":.metadata.namespace)"
```

2. Obtenir l'état des pods **virt-operator**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-operator
```

3. Consultez les journaux du pod **virt-operator** pour déterminer l'état du leader :

```
$ oc -n $NAMESPACE logs | grep lead
```

Exemple d'un pod leader :

```
{"component":"virt-operator","level":"info","msg":"Attempting to acquire
leader status","pos":"application.go:400","timestamp":"2021-11-30T12:15:18.635387Z"}
I1130 12:15:18.635452       1 leaderelection.go:243] attempting to acquire
leader lease <namespace>/virt-operator...
I1130 12:15:19.216582       1 leaderelection.go:253] successfully acquired
lease <namespace>/virt-operator
{"component":"virt-operator","level":"info","msg":"Started leading",
"pos":"application.go:385","timestamp":"2021-11-30T12:15:19.216836Z"}
```

Exemple de pod non leader :

```
{"component":"virt-operator","level":"info","msg":"Attempting to acquire
leader status","pos":"application.go:400","timestamp":"2021-11-30T12:15:20.533696Z"}
I1130 12:15:20.533792       1 leaderelection.go:243] attempting to acquire
leader lease <namespace>/virt-operator...
```

4. Obtenir les détails des pods **virt-operator** concernés :

```
oc -n $NAMESPACE describe pod <virt-operator>
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez de trouver la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.30. NoReadyVirtController

Signification
Cette alerte se déclenche lorsqu'aucun dispositif **virt-controller** n'a été détecté pendant 5 minutes.

Les dispositifs **virt-controller** contrôlent les définitions de ressources personnalisées des instances de machines virtuelles (VMI) et gèrent les pods associés. Les dispositifs créent des pods pour les VMI et gèrent le cycle de vie des pods.

Par conséquent, les dispositifs **virt-controller** sont essentiels pour toutes les fonctionnalités de virtualisation à l'échelle de la grappe.

Impact
Toutes les actions liées à la gestion du cycle de vie des machines virtuelles échouent. Il s'agit notamment du lancement d'une nouvelle VMI ou de l'arrêt d'une VMI existante.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez le nombre d'appareils **virt-controller**:

```
$ oc get deployment -n $NAMESPACE virt-controller \
  -o jsonpath='{.status.readyReplicas}'
```

3. Vérifiez l'état du déploiement de **virt-controller**:

```
$ oc -n $NAMESPACE get deploy virt-controller -o yaml
```

4. Obtenez les détails du déploiement de **virt-controller** pour vérifier les conditions d'état telles que les pods en panne ou l'impossibilité d'extraire des images :

```
$ oc -n $NAMESPACE describe deploy virt-controller
```

5. Obtenir les informations sur les nacelles du site **virt-controller**:

```
$ get pods -n $NAMESPACE | grep virt-controller
```

6. Vérifiez les journaux des pods **virt-controller** pour les messages d'erreur :

```
oc logs -n $NAMESPACE <virt-controller>
```

7. Vérifiez que les nœuds ne présentent pas de problèmes, tels que l'état **NotReady**:

```
$ oc get nodes
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez de trouver la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.31. NoReadyVirtOperator

Signification
Cette alerte se déclenche lorsqu'aucun pod **virt-operator** en état **Ready** n'a été détecté pendant 10 minutes.

Le site **virt-operator** est le premier opérateur à démarrer dans une grappe. Ses principales responsabilités sont les suivantes

- Installation, mise à jour en direct et mise à niveau en direct d'un cluster
- Surveiller le cycle de vie des contrôleurs de premier niveau, tels que **virt-controller**, **virt-handler**, **virt-launcher**, et gérer leur rapprochement
- Certaines tâches à l'échelle de la grappe, telles que la rotation des certificats et la gestion de l'infrastructure

Le déploiement par défaut est constitué de deux pods **virt-operator**.

Impact
Cette alerte indique une défaillance au niveau du cluster. Les fonctionnalités essentielles de gestion des clusters, telles que la rotation des certifications, la mise à niveau et le rapprochement des contrôleurs, risquent de ne pas être disponibles.

Le site **virt-operator** n'est pas directement responsable des machines virtuelles dans le cluster. Par conséquent, son indisponibilité temporaire n'affecte pas de manière significative les charges de travail.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Obtenir le nom du déploiement **virt-operator**:

```
$ oc -n $NAMESPACE get deploy virt-operator -o yaml
```

3. Générer la description du déploiement de **virt-operator**:

```
$ oc -n $NAMESPACE describe deploy virt-operator
```

4. Vérifier l'absence de problèmes au niveau du nœud, tels que l'état de **NotReady**:

```
$ oc get nodes
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les artefacts recueillis au cours de la procédure de diagnostic.

14.14.32. OrphanedVirtualMachineInstances (substances de machines virtuelles orphelines)

Signification
Cette alerte se déclenche lorsqu'une instance de machine virtuelle (VMI), ou un pod **virt-launcher**, s'exécute sur un nœud qui n'a pas de pod **virt-handler** en cours d'exécution. Une telle VMI est appelée *orphaned*.

Impact
Les IMV orphelines ne peuvent pas être gérées.

Diagnostic

1. Vérifiez l'état des pods **virt-handler** pour connaître les nœuds sur lesquels ils s'exécutent :

```
$ oc get pods --all-namespaces -o wide -l kubevirt.io=virt-handler
```

2. Vérifiez l'état des IMV pour identifier les IMV en cours d'exécution sur des nœuds qui n'ont pas de pod **virt-handler** en cours d'exécution :

```
$ oc get vmis --all-namespaces
```

3. Vérifiez l'état du démon **virt-handler**:

```
$ oc get daemonset virt-handler --all-namespaces
```

Exemple de sortie

```
NAME         DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE ...
virt-handler 2        2        2      2           2         ...
```

Le jeu de démons est considéré comme sain si les colonnes **Desired**, **Ready**, et **Available** contiennent la même valeur.

4. Si le jeu de démons **virt-handler** n'est pas sain, vérifiez si le jeu de démons **virt-handler** ne présente pas de problèmes de déploiement de pods :

```
$ oc get daemonset virt-handler --all-namespaces -o yaml | jq .status
```

5. Vérifiez que les nœuds ne présentent pas de problèmes tels que l'état **NotReady**:

```
$ oc get nodes
```

6. Vérifiez la strophe **spec.workloads** de la ressource personnalisée (CR) **KubeVirt** pour une politique de placement des charges de travail :

```
$ oc get kubevirt kubevirt --all-namespaces -o yaml
```

Atténuation
Si une stratégie de placement des charges de travail est configurée, ajoutez le nœud avec la VMI à la stratégie.

Les causes possibles de la suppression d'un pod **virt-handler** d'un nœud sont notamment les modifications apportées aux taches et tolérances du nœud ou aux règles de programmation d'un pod.

Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.33. Charges de travail d'instances de machines virtuelles périmées

Signification
Cette alerte se déclenche lorsque des instances de machines virtuelles (VMI) en cours d'exécution dans des pods **virt-launcher** obsolètes sont détectées 24 heures après la mise à jour du plan de contrôle d'OpenShift Virtualization.

Impact
Les IMV obsolètes peuvent ne pas avoir accès aux nouvelles fonctionnalités d'OpenShift Virtualization.

Les IMV obsolètes ne recevront pas les correctifs de sécurité associés à la mise à jour du pod **virt-launcher**.

Diagnostic

1. Identifier les IMV périmés :

```
$ oc get vmi -l kubevirt.io/outdatedLauncherImage --all-namespaces
```

2. Vérifiez la ressource personnalisée (CR) **KubeVirt** pour déterminer si **workloadUpdateMethods** est configuré dans la strophe **workloadUpdateStrategy**:

```
$ oc get kubevirt kubevirt --all-namespaces -o yaml
```

3. Vérifier chaque IMV obsolète pour déterminer s'il est possible de le migrer en direct :

```
$ oc get vmi <vmi> -o yaml
```

Exemple de sortie

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstance
...
  status:
    conditions:
    - lastProbeTime: null
      lastTransitionTime: null
      message: cannot migrate VMI which does not use masquerade
      to connect to the pod network
      reason: InterfaceNotLiveMigratable
      status: "False"
      type: LiveMigratable
```

Atténuation
Configuration des mises à jour automatisées de la charge de travail
Mettre à jour le CR **HyperConverged** pour activer les mises à jour automatiques de la charge de travail.

Arrêt d'une VM associée à une VMI non migrable en direct

- Si une IMV n'est pas migrable en direct et si **runStrategy: always** est défini dans l'objet **VirtualMachine** correspondant, vous pouvez mettre à jour l'IMV en arrêtant manuellement la machine virtuelle (VM) :

```
virctl stop --namespace <namespace> <vm>
```

Un nouveau VMI démarre immédiatement dans un pod **virt-launcher** mis à jour pour remplacer le VMI arrêté. C'est l'équivalent d'une action de redémarrage.

NOTE

L'arrêt manuel d'une VM *live-migratable* est destructeur et déconseillé car il interrompt la charge de travail.

Migration d'une VMI migrable en direct

Si une IMV est migrable en direct, vous pouvez la mettre à jour en créant un objet **VirtualMachineInstanceMigration** qui cible une IMV spécifique en cours d'exécution. L'IMV est migrée dans un pod **virt-launcher** mis à jour.

1. Créez un manifeste **VirtualMachineInstanceMigration** et enregistrez-le sous **migration.yaml**:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstanceMigration
metadata:
  name: <migration_name>
  namespace: <namespace>
spec:
  vmiName: <vmi_name>
```

2. Créer un objet **VirtualMachineInstanceMigration** pour déclencher la migration :

```
$ oc create -f migration.yaml
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.34. SSPCommonTemplatesModificationReverted

Signification
Cette alerte se déclenche lorsque l'opérateur SSP (Scheduling, Scale, and Performance) annule les modifications apportées aux modèles communs dans le cadre de sa procédure de réconciliation.

L'opérateur SSP déploie et réconcilie les modèles communs et le validateur de modèle. Si un utilisateur ou un script modifie un modèle commun, les modifications sont annulées par l'opérateur SSP.

Impact
Les modifications apportées aux modèles communs sont écrasées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | grep ssp-operator | \
  awk '{print $1}')"
```

2. Vérifiez les journaux de **ssp-operator** pour les modèles dont les modifications ont été annulées :

```
$ oc -n $NAMESPACE logs --tail=-1 -l control-plane=ssp-operator | \
  grep 'common template' -C 3
```

Atténuation
Essayez d'identifier et de résoudre la cause des changements.

Veillez à ce que les modifications ne soient apportées qu'aux copies des modèles, et non aux modèles eux-mêmes.

14.14.35. SSPFailingToReconcile (défaut de réconciliation)

Signification

Cette alerte se déclenche lorsque le cycle de rapprochement de l'opérateur SSP (Scheduling, Scale and Performance) échoue à plusieurs reprises, bien que l'opérateur SSP soit en cours d'exécution.

L'opérateur SSP est responsable du déploiement et du rapprochement des modèles communs et du validateur de modèles.

Impact
Les composants dépendants peuvent ne pas être déployés. Les modifications apportées aux composants peuvent ne pas être rapprochées. Par conséquent, les modèles communs ou le validateur de modèles peuvent ne pas être mis à jour ou réinitialisés en cas d'échec.

Diagnostic

1. Exporter la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | grep ssp-operator | \
  awk '{print $1}')"
```

2. Obtenir les informations sur les nacelles du site **ssp-operator**:

```
$ oc -n $NAMESPACE describe pods -l control-plane=ssp-operator
```

3. Vérifiez si des erreurs se sont produites sur le site **ssp-operator**:

```
$ oc -n $NAMESPACE logs --tail=-1 -l control-plane=ssp-operator
```

4. Obtenir l'état des pods **virt-template-validator**:

```
$ oc -n $NAMESPACE get pods -l name=virt-template-validator
```

5. Obtenir les informations sur les nacelles du site **virt-template-validator**:

```
$ oc -n $NAMESPACE describe pods -l name=virt-template-validator
```

6. Vérifiez si des erreurs se sont produites sur le site **virt-template-validator**:

```
$ oc -n $NAMESPACE logs --tail=-1 -l name=virt-template-validator
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.36. SSPHighRateRejectedVms

Signification
Cette alerte se déclenche lorsqu'un utilisateur ou un script tente de créer ou de modifier un grand nombre de machines virtuelles (VM) à l'aide d'une configuration non valide.

Impact
Les machines virtuelles ne sont ni créées ni modifiées. Par conséquent, l'environnement peut ne pas se comporter comme prévu.

Diagnostic

1. Exporter la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | grep ssp-operator | \
  awk '{print $1}')"
```

2. Vérifiez les journaux de **virt-template-validator** pour les erreurs qui pourraient indiquer la cause :

```
$ oc -n $NAMESPACE logs --tail=-1 -l name=virt-template-validator
```

Exemple de sortie

```
{"component":"kubevirt-template-validator","level":"info","msg":"evalution
summary for ubuntu-3166wmdbbfkroku0:\nminimal-required-memory applied: FAIL,
value 1073741824 is lower than minimum [2147483648]\n\nsucceeded=false",
"pos":"admission.go:25","timestamp":"2021-09-28T17:59:10.934470Z"}
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.37. SSPOperatorDown

Signification
Cette alerte se déclenche lorsque tous les pods de l'opérateur SSP (Scheduling, Scale and Performance) sont hors service.

L'opérateur SSP est responsable du déploiement et du rapprochement des modèles communs et du validateur de modèles.

Impact
Les composants dépendants peuvent ne pas être déployés. Les modifications apportées aux composants peuvent ne pas être rapprochées. Par conséquent, les modèles communs et/ou le validateur de modèles peuvent ne pas être mis à jour ou réinitialisés en cas d'échec.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | grep ssp-operator | \
  awk '{print $1}')"
```

2. Vérifier l'état des pods **ssp-operator**.

```
$ oc -n $NAMESPACE get pods -l control-plane=ssp-operator
```

3. Obtenir les informations sur les nacelles du site **ssp-operator**:

```
$ oc -n $NAMESPACE describe pods -l control-plane=ssp-operator
```

4. Vérifiez les messages d'erreur dans les journaux de **ssp-operator**:

```
$ oc -n $NAMESPACE logs --tail=-1 -l control-plane=ssp-operator
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.38. SSPTemplateValidatorDown

Signification
Cette alerte se déclenche lorsque tous les modules du validateur de modèle sont hors service.

Le validateur de modèles vérifie les machines virtuelles (VM) pour s'assurer qu'elles ne violent pas leurs modèles.

Impact
Les machines virtuelles ne sont pas validées par rapport à leurs modèles. Par conséquent, les machines virtuelles peuvent être créées avec des spécifications qui ne correspondent pas à leurs charges de travail respectives.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get deployment -A | grep ssp-operator | \
  awk '{print $1}')"
```

2. Obtenir l'état des pods **virt-template-validator**:

```
$ oc -n $NAMESPACE get pods -l name=virt-template-validator
```

3. Obtenir les informations sur les nacelles du site **virt-template-validator**:

```
$ oc -n $NAMESPACE describe pods -l name=virt-template-validator
```

4. Vérifiez les messages d'erreur dans les journaux de **virt-template-validator**:

```
$ oc -n $NAMESPACE logs --tail=-1 -l name=virt-template-validator
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.39. VirtAPIDown

Signification
Cette alerte se déclenche lorsque tous les pods du serveur API sont hors service.

Impact

Les objets OpenShift Virtualization ne peuvent pas envoyer d'appels API.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état des pods **virt-api**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-api
```

3. Vérifiez l'état du déploiement de **virt-api**:

```
$ oc -n $NAMESPACE get deploy virt-api -o yaml
```

4. Vérifiez les détails du déploiement de **virt-api** pour des problèmes tels que des pods en panne ou des échecs d'extraction d'image :

```
$ oc -n $NAMESPACE describe deploy virt-api
```

5. Vérifiez si des nœuds se trouvent dans l'état **NotReady**:

```
$ oc get nodes
```

Atténuation
Essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.40. VirtApiRESTErrorsBurst

Signification
Plus de 80 % des appels REST ont échoué dans les pods **virt-api** au cours des 5 dernières minutes.

Impact
Un taux très élevé d'échecs des appels REST à **virt-api** peut entraîner une lenteur de réponse et d'exécution des appels d'API, voire un rejet total des appels d'API.

Toutefois, les charges de travail des machines virtuelles en cours d'exécution ne devraient pas être affectées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Obtenez la liste des pods **virt-api** sur votre déploiement :

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-api
```

3. Vérifiez les messages d'erreur dans les journaux de **virt-api**:

```
oc logs -n $NAMESPACE <virt-api>
```

4. Obtenir les informations sur les nacelles du site **virt-api**:

```
oc describe -n $NAMESPACE <virt-api>
```

5. Vérifiez si des problèmes sont survenus avec les nœuds. Par exemple, ils peuvent être dans l'état **NotReady**:

```
$ oc get nodes
```

6. Vérifiez l'état du déploiement de **virt-api**:

```
$ oc -n $NAMESPACE get deploy virt-api -o yaml
```

7. Obtenir les détails du déploiement de **virt-api**:

```
$ oc -n $NAMESPACE describe deploy virt-api
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.41. VirtApiRESTErrorsHigh

Signification
Plus de 5 % des appels REST ont échoué dans les pods **virt-api** au cours des 60 dernières minutes.

Impact
Un taux élevé d'échecs des appels REST à **virt-api** peut entraîner un ralentissement de la réponse et de l'exécution des appels API.

Toutefois, les charges de travail des machines virtuelles en cours d'exécution ne devraient pas être affectées.

Diagnostic

1. Définissez la variable d'environnement **NAMESPACE** comme suit :

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état des pods **virt-api**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-api
```

3. Consultez les journaux de **virt-api**:

```
$ oc logs -n  $NAMESPACE <virt-api>
```

4. Obtenir les informations sur les nacelles du site **virt-api**:

```
oc describe -n $NAMESPACE <virt-api>
```

5. Vérifiez si des problèmes sont survenus avec les nœuds. Par exemple, ils peuvent être dans l'état **NotReady**:

```
$ oc get nodes
```

6. Vérifiez l'état du déploiement de **virt-api**:

```
$ oc -n $NAMESPACE get deploy virt-api -o yaml
```

7. Obtenir les détails du déploiement de **virt-api**:

```
$ oc -n $NAMESPACE describe deploy virt-api
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez d'identifier la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.42. VirtControllerDown

Signification
Aucun pod **virt-controller** en cours de fonctionnement n'a été détecté pendant 5 minutes.

Impact
Toutes les actions liées à la gestion du cycle de vie des machines virtuelles (VM) échouent. Il s'agit notamment du lancement d'une nouvelle instance de machine virtuelle (VMI) ou de l'arrêt d'une VMI existante.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez l'état du déploiement de **virt-controller**:

```
$ oc get deployment -n $NAMESPACE virt-controller -o yaml
```

3. Examinez les journaux du pod **virt-controller**:

```
$ oc get logs <virt-controller>
```

Atténuation
Cette alerte peut avoir diverses causes, dont les suivantes :

- Épuisement des ressources du nœud
- Mémoire insuffisante sur le cluster
- Les nœuds sont hors service
- Le serveur API est surchargé. Par exemple, le planificateur peut être très sollicité et n'est donc pas entièrement disponible.
- Questions relatives à la mise en réseau

Identifiez la cause première et corrigez-la, si possible.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.43. VirtControllerRESTErrorsBurst

Signification
Plus de 80 % des appels REST dans les pods **virt-controller** ont échoué au cours des 5 dernières minutes.

Le site **virt-controller** a probablement perdu complètement la connexion avec le serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-controller** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les mises à jour d'état ne sont pas propagées et les actions telles que les migrations ne peuvent pas avoir lieu. Cependant, les charges de travail en cours d'exécution ne sont pas affectées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Liste des pods **virt-controller** disponibles :

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-controller
```

3. Vérifiez dans les journaux de **virt-controller** les messages d'erreur lors de la connexion au serveur API :

```
$ oc logs -n  $NAMESPACE <virt-controller>
```

Atténuation

- Si le module **virt-controller** ne peut pas se connecter au serveur API, supprimez le module pour

forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-controller>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.44. VirtControllerRESTErrorsHigh

Signification
Plus de 5 % des appels REST ont échoué sur **virt-controller** au cours des 60 dernières minutes.

Cela est probablement dû au fait que **virt-controller** a partiellement perdu la connexion avec le serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-controller** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les actions liées aux nœuds, telles que le démarrage, la migration et la planification des machines virtuelles, sont retardées. Les charges de travail en cours d'exécution ne sont pas affectées, mais la communication de leur état actuel peut être retardée.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Liste des pods **virt-controller** disponibles :

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-controller
```

3. Vérifiez dans les journaux de **virt-controller** les messages d'erreur lors de la connexion au serveur API :

```
$ oc logs -n  $NAMESPACE <virt-controller>
```

Atténuation

- Si le module **virt-controller** ne peut pas se connecter au serveur API, supprimez le module pour forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-controller>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.45. VirtHandlerDaemonSetRolloutFailing

Signification
Le jeu de démons **virt-handler** n'a pas réussi à se déployer sur un ou plusieurs nœuds de travail après 15 minutes.

Impact
Cette alerte est un avertissement. Elle n'indique pas que tous les ensembles de démons **virt-handler** n'ont pas été déployés. Par conséquent, le cycle de vie normal des machines virtuelles n'est pas affecté, sauf si le cluster est surchargé.

Diagnostic
Identifiez les nœuds de travail qui n'ont pas de pod **virt-handler** en cours d'exécution :

1. Exporter la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez l'état des pods **virt-handler** pour identifier les pods qui n'ont pas été déployés :

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-handler
```

3. Obtenir le nom du nœud de travail du pod **virt-handler**:

```
oc -n $NAMESPACE get pod <virt-handler> -o jsonpath='{.spec.nodeName}'
```

Atténuation
Si les pods **virt-handler** n'ont pas pu être déployés en raison de ressources insuffisantes, vous pouvez supprimer d'autres pods sur le nœud de travail affecté.

14.14.46. VirtHandlerRESTErrorsBurst

Signification
Plus de 80 % des appels REST ont échoué sur **virt-handler** au cours des 5 dernières minutes. Cette alerte indique généralement que les pods **virt-handler** ne peuvent pas se connecter au serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-handler** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les mises à jour d'état ne sont pas propagées et les actions liées au nœud, telles que les migrations, échouent. Cependant, les charges de travail en cours d'exécution sur le nœud affecté ne sont pas affectées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état du pod **virt-handler**:

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-handler
```

3. Vérifiez dans les journaux de **virt-handler** les messages d'erreur lors de la connexion au serveur API :

```
$ oc logs -n  $NAMESPACE <virt-handler>
```

Atténuation

- Si le site **virt-handler** ne peut pas se connecter au serveur API, supprimez le module pour forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-handler>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.47. VirtHandlerRESTErrorsHigh

Signification
Plus de 5 % des appels REST ont échoué sur **virt-handler** au cours des 60 dernières minutes. Cette alerte indique généralement que les pods **virt-handler** ont partiellement perdu la connexion au serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-handler** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les actions liées aux nœuds, telles que le démarrage et la migration des charges de travail, sont retardées sur le nœud sur lequel **virt-handler** est exécuté. Les charges de travail en cours d'exécution ne sont pas affectées, mais la communication de leur état actuel peut être retardée.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état du pod **virt-handler**:

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-handler
```

3. Vérifiez dans les journaux de **virt-handler** les messages d'erreur lors de la connexion au serveur API :

```
$ oc logs -n $NAMESPACE <virt-handler>
oc logs -n $NAMESPACE <virt-handler>
```

Atténuation

- Si le site **virt-handler** ne peut pas se connecter au serveur API, supprimez le module pour forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-handler>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.48. VirtOperatorDown

Signification
Cette alerte se déclenche lorsqu'aucun pod **virt-operator** dans l'état **Running** n'a été détecté pendant 10 minutes.

Le site **virt-operator** est le premier opérateur à démarrer dans une grappe. Ses principales responsabilités sont les suivantes

- Installation, mise à jour en direct et mise à niveau en direct d'un cluster
- Surveiller le cycle de vie des contrôleurs de premier niveau, tels que **virt-controller**, **virt-handler**, **virt-launcher**, et gérer leur rapprochement
- Certaines tâches à l'échelle de la grappe, telles que la rotation des certificats et la gestion de l'infrastructure

Le déploiement **virt-operator** a une réplique par défaut de 2 pods.

Impact
Cette alerte indique une défaillance au niveau de la grappe. Les fonctionnalités critiques de gestion à l'échelle de la grappe, telles que la rotation des certifications, la mise à niveau et le rapprochement des contrôleurs, peuvent ne pas être disponibles.

Le site **virt-operator** n'est pas directement responsable des machines virtuelles (VM) dans le cluster. Par conséquent, son indisponibilité temporaire n'affecte pas de manière significative les charges de travail des machines virtuelles.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifiez l'état du déploiement de **virt-operator**:

```
$ oc -n $NAMESPACE get deploy virt-operator -o yaml
```

3. Obtenir les détails du déploiement de **virt-operator**:

```
$ oc -n $NAMESPACE describe deploy virt-operator
```

4. Vérifier l'état des pods **virt-operator**:

```
$ oc get pods -n $NAMESPACE -l=kubevirt.io=virt-operator
```

5. Vérifier l'absence de problèmes au niveau du nœud, tels que l'état de **NotReady**:

```
$ oc get nodes
```

Atténuation
Sur la base des informations obtenues au cours de la procédure de diagnostic, essayez de trouver la cause première et de résoudre le problème.

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.49. VirtOperatorRESTErrorsBurst

Signification
Cette alerte se déclenche lorsque plus de 80 % des appels REST dans les modules **virt-operator** ont échoué au cours des 5 dernières minutes. Cela indique généralement que les modules **virt-operator** ne peuvent pas se connecter au serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-operator** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les actions au niveau du cluster, telles que la mise à niveau et le rapprochement des contrôleurs, peuvent ne pas être disponibles.

Toutefois, les charges de travail telles que les machines virtuelles (VM) et les instances VM (VMI) ne devraient pas être affectées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état des pods **virt-operator**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-operator
```

3. Vérifiez dans les journaux de **virt-operator** les messages d'erreur lors de la connexion au serveur API :

```
oc -n $NAMESPACE logs <virt-operator>
```

4. Obtenir les coordonnées de la nacelle **virt-operator**:

```
oc -n $NAMESPACE describe pod <virt-operator>
```

Atténuation

- Si le module **virt-operator** ne peut pas se connecter au serveur API, supprimez le module pour forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-operator>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.50. VirtOperatorRESTErrorsHigh

Signification
Cette alerte se déclenche lorsque plus de 5 % des appels REST dans les pods **virt-operator** ont échoué au cours des 60 dernières minutes. Cela indique généralement que les modules **virt-operator** ne peuvent pas se connecter au serveur API.

Cette erreur est souvent due à l'un des problèmes suivants :

- Le serveur API est surchargé, ce qui entraîne des dépassements de délai. Pour vérifier si c'est le cas, vérifiez les métriques du serveur API et affichez ses temps de réponse et le nombre total d'appels.
- Le pod **virt-operator** ne peut pas atteindre le serveur API. Cela est généralement dû à des problèmes de DNS sur le nœud et à des problèmes de connectivité réseau.

Impact
Les actions au niveau du cluster, telles que la mise à niveau et le rapprochement des contrôleurs, peuvent être retardées.

Toutefois, les charges de travail telles que les machines virtuelles (VM) et les instances VM (VMI) ne devraient pas être affectées.

Diagnostic

1. Définir la variable d'environnement **NAMESPACE**:

```
$ export NAMESPACE="$(oc get kubevirt -A \
  -o custom-columns="":.metadata.namespace)"
```

2. Vérifier l'état des pods **virt-operator**:

```
$ oc -n $NAMESPACE get pods -l kubevirt.io=virt-operator
```

3. Vérifiez dans les journaux de **virt-operator** les messages d'erreur lors de la connexion au serveur API :

```
oc -n $NAMESPACE logs <virt-operator>
```

4. Obtenir les coordonnées de la nacelle **virt-operator**:

```
oc -n $NAMESPACE describe pod <virt-operator>
```

Atténuation

- Si le module **virt-operator** ne peut pas se connecter au serveur API, supprimez le module pour forcer un redémarrage :

```
oc delete -n $NAMESPACE <virt-operator>
```

Si vous ne parvenez pas à résoudre le problème, connectez-vous au portail client et ouvrez un dossier d'assistance, en joignant les éléments recueillis au cours de la procédure de diagnostic.

14.14.51. VMCannotBeEvicted

Signification
Cette alerte se déclenche lorsque la stratégie d'éviction d'une machine virtuelle (VM) est définie sur **LiveMigration** mais que la VM n'est pas migrable.

Impact
Les machines virtuelles non migrables empêchent l'éviction des nœuds. Cette condition affecte les opérations telles que la vidange et la mise à jour des nœuds.

Diagnostic

1. Vérifier la configuration VMI pour déterminer si la valeur de **evictionStrategy** est **LiveMigrate**:

```
$ oc get vmis -o yaml
```

2. Vérifiez l'état **False** dans la colonne **LIVE-MIGRATABLE** pour identifier les IMV qui ne sont pas migrables :

```
$ oc get vmis -o wide
```

3. Obtenez les détails de l'IMV et consultez le site **spec.conditions** pour identifier le problème :

```
$ oc get vmi <vmi> -o yaml
```

Exemple de sortie

```
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: null
    message: cannot migrate VMI which does not use masquerade to connect
    to the pod network
    reason: InterfaceNotLiveMigratable
    status: "False"
    type: LiveMigratable
```

Atténuation
Réglez le site **evictionStrategy** de la VMI sur **shutdown** ou résolvez le problème qui empêche la VMI de migrer.

14.15. COLLECTE DE DONNÉES POUR RED HAT SUPPORT

Lorsque vous soumettez un cas d'assistance à Red Hat Support, il est utile de fournir des informations de débogage pour OpenShift Container Platform et OpenShift Virtualization en utilisant les outils suivants :

outil indispensable

L'outil **must-gather** recueille des informations de diagnostic, notamment des définitions de ressources et des journaux de service.

Prometheus

Prometheus est une base de données de séries temporelles et un moteur d'évaluation de règles pour les métriques. Prometheus envoie des alertes à Alertmanager pour traitement.

Gestionnaire d'alerte

Le service Alertmanager gère les alertes reçues de Prometheus. Il est également chargé d'envoyer les alertes aux systèmes de notification externes.

14.15.1. Collecte de données sur votre environnement

La collecte de données sur votre environnement réduit le temps nécessaire à l'analyse et à la détermination de la cause première.

Conditions préalables

- Fixez la durée de conservation des données de métrologie Prometheus à un minimum de sept jours.
- Configurer l'Alertmanager pour qu'il capture les alertes pertinentes et les envoie à une boîte aux lettres dédiée afin qu'elles puissent être consultées et conservées en dehors du cluster.
- Enregistrez le nombre exact de nœuds et de machines virtuelles affectés.

Procédure

1. Collectez les données **must-gather** pour la grappe en utilisant l'image par défaut **must-gather**.
2. Recueillir les données **must-gather** pour Red Hat OpenShift Data Foundation, si nécessaire.
3. Collectez les données **must-gather** pour OpenShift Virtualization en utilisant l'image OpenShift Virtualization **must-gather**.
4. Collecter les métriques Prometheus pour le cluster.

14.15.1.1. Ressources supplémentaires

- Configuration de la durée de conservation des données de métrologie de Prometheus
- Configuration de l'Alertmanager pour l'envoi de notifications d'alerte à des systèmes externes
- Collecte des données **must-gather** pour OpenShift Container Platform

- Collecte des données **must-gather** pour Red Hat OpenShift Data Foundation
- Collecte des données **must-gather** pour OpenShift Virtualization
- Collecte des métriques Prometheus pour tous les projets en tant qu'administrateur de cluster

14.15.2. Collecte de données sur les machines virtuelles

La collecte de données sur les machines virtuelles (VM) qui fonctionnent mal réduit le temps nécessaire à l'analyse et à la détermination de la cause première.

Conditions préalables

- VM Windows :
  - Enregistrez les détails de la mise à jour des correctifs Windows pour Red Hat Support.
  - Installer la dernière version des pilotes VirtIO. Les pilotes VirtIO incluent l'agent invité QEMU.
  - Si le protocole Remote Desktop Protocol (RDP) est activé, essayez de vous connecter aux machines virtuelles avec RDP pour déterminer s'il y a un problème avec le logiciel de connexion.

Procédure

1. Recueillir des données détaillées sur **must-gather** concernant les machines virtuelles défectueuses.
2. Réalisez des captures d'écran des machines virtuelles qui se sont bloquées avant de les redémarrer.
3. Enregistrez les facteurs que les machines virtuelles défectueuses ont en commun. Par exemple, les VM ont le même hôte ou le même réseau.

14.15.2.1. Ressources supplémentaires

- Installation des pilotes VirtIO sur les VM Windows
- Téléchargement et installation des pilotes VirtIO sur les VM Windows sans accès à l'hôte
- Connexion aux machines virtuelles Windows avec RDP en utilisant la console web ou la ligne de commande
- Collecte de données **must-gather** sur les machines virtuelles

14.15.3. Utilisation de l'outil must-gather pour OpenShift Virtualization

Vous pouvez collecter des données sur les ressources d'OpenShift Virtualization en exécutant la commande **must-gather** avec l'image d'OpenShift Virtualization.

La collecte de données par défaut comprend des informations sur les ressources suivantes :

- Espaces de noms de l'opérateur de virtualisation OpenShift, y compris les objets enfants
- Définitions de ressources personnalisées pour OpenShift Virtualization

- Espaces de noms contenant des machines virtuelles
- Définitions de base des machines virtuelles

Procédure

- Exécutez la commande suivante pour collecter des données sur OpenShift Virtualization :

```
$ oc adm must-gather --image-stream=openshift/must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-
rhel8:v{HCOVersion}
```

14.15.3.1. options d'outils indispensables

Vous pouvez spécifier une combinaison de scripts et de variables d'environnement pour les options suivantes :

- Collecte d'informations détaillées sur les machines virtuelles (VM) à partir d'un espace de noms
- Collecte d'informations détaillées sur les machines virtuelles spécifiées
- Collecte d'informations sur les images et les flux d'images
- Limiter le nombre maximum de processus parallèles utilisés par l'outil **must-gather**

14.15.3.1.1. Parameters

Variables d'environnement

Vous pouvez spécifier des variables d'environnement pour un script compatible.

NS=<namespace_name>

Collecte des informations sur les machines virtuelles, y compris les détails du pod **virt-launcher**, à partir de l'espace de noms que vous avez spécifié. Les données CR **VirtualMachine** et **VirtualMachineInstance** sont collectées pour tous les espaces de noms.

VM=<vm_name>

Collecter des informations sur une machine virtuelle particulière. Pour utiliser cette option, vous devez également spécifier un espace de noms en utilisant la variable d'environnement **NS**.

PROS=<number_of_processes>

Modifier le nombre maximum de processus parallèles utilisés par l'outil **must-gather**. La valeur par défaut est **5**.

IMPORTANT

L'utilisation d'un trop grand nombre de processus parallèles peut entraîner des problèmes de performance. Il n'est pas recommandé d'augmenter le nombre maximal de processus parallèles.

Scripts

Chaque script n'est compatible qu'avec certaines combinaisons de variables d'environnement.

gather_vms_details

Collecter les fichiers journaux de VM, les définitions de VM et les espaces de noms (et leurs objets enfants) qui appartiennent aux ressources OpenShift Virtualization. Si vous utilisez ce paramètre sans spécifier d'espace de noms ou de VM, l'outil **must-gather** collecte ces données pour toutes les VM du cluster. Ce script est compatible avec toutes les variables d'environnement, mais vous devez spécifier un espace de noms si vous utilisez la variable **VM**.

gather

Utilisez le script par défaut **must-gather**, qui collecte les données de cluster de tous les espaces de noms et n'inclut que des informations de base sur les machines virtuelles. Ce script n'est compatible qu'avec la variable **PROS**.

gather_images

Collecte des informations sur les ressources personnalisées des images et des flux d'images. Ce script n'est compatible qu'avec la variable **PROS**.

14.15.3.1.2. Utilisation et exemples

Les variables d'environnement sont facultatives. Vous pouvez exécuter un script seul ou avec une ou plusieurs variables d'environnement compatibles.

Tableau 14.1. Paramètres de compatibilité

Le scénario	Variable d'environnement compatible
gather_vms_details	• Pour un espace de noms : **NS=<namespace_name>** • Pour une VM : **VM=<vm_name> NS=<namespace_name>** • **PROS=<number_of_processes>**
gather	• **PROS=<number_of_processes>**
gather_images	• **PROS=<number_of_processes>**

Pour personnaliser les données collectées par **must-gather**, vous devez ajouter un double tiret (**--**) à la commande, suivi d'un espace et d'un ou plusieurs paramètres compatibles.

Syntaxe

```
$ oc adm must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v4.12.2 \
  -- <environment_variable_1> <environment_variable_2> <script_name>
```

Informations détaillées sur les machines virtuelles

La commande suivante recueille des informations détaillées sur la VM **my-vm** dans l'espace de noms **mynamespace**:

-

```
$ oc adm must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v4.12.2 \
  -- NS=mynamespace VM=my-vm gather_vms_details 1
```

1 La variable d'environnement **NS** est obligatoire si vous utilisez la variable d'environnement **VM**.

Collecte de données par défaut limitée à trois processus parallèles

La commande suivante recueille des informations sur le site **must-gather** par défaut en utilisant un maximum de trois processus parallèles :

```
$ oc adm must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v4.12.2 \
  -- PROS=3 gather
```

Informations sur les images et les flux d'images

La commande suivante permet de collecter des informations sur les images et les flux d'images de la grappe :

```
$ oc adm must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v4.12.2 \
  -- gather_images
```

14.15.3.2. Ressources supplémentaires

- À propos de l'outil **must-gather**

CHAPITRE 15. SAUVEGARDE ET RESTAURATION

15.1. INSTALLATION ET CONFIGURATION DE L'OADP

En tant qu'administrateur de cluster, vous installez l'API OpenShift pour la protection des données (OADP) en installant l'opérateur OADP. L'opérateur installe Velero 1.9.

Vous créez une adresse **Secret** par défaut pour votre fournisseur de stockage de sauvegarde, puis vous installez l'application de protection des données.

15.1.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec les privilèges **cluster-admin**.

Procédure

1. Dans la console Web OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.
3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

15.1.2. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.

IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

15.1.2.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

15.1.3. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

15.1.3.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Spécifier le sélecteur de nœud à fournir à Velero podSpec

15.1.3.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

1. Indiquez la chaîne du certificat de l'autorité de certification codée en base64.
2. La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

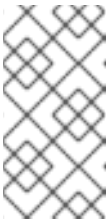
15.1.4. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.
- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).
- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials**.
- Si les emplacements de sauvegarde et de cliché utilisent des informations d'identification différentes, vous devez créer deux sites **Secrets**:

- **Secret** avec un nom personnalisé pour l'emplacement de la sauvegarde. Vous ajoutez ce **Secret** au CR **DataProtectionApplication**.
- **Secret** avec le nom par défaut, **cloud-credentials**, pour l'emplacement de l'instantané. Ce site **Secret** n'est pas référencé dans le CR **DataProtectionApplication**.



NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (1)
        - gcp (2)
        - csi (3)
        - openshift (4)
    restic:
      enable: true (5)
      podConfig:
        nodeSelector: <node selector> (6)
  backupLocations:
    - velero:
        provider: gcp (7)
        default: true
        credential:
          key: cloud
          name: <default_secret> (8)
        objectStorage:
          bucket: <bucket_name> (9)
          prefix: <prefix> (10)
```

(1) Le plugin **kubevirt** est obligatoire pour OpenShift Virtualization.

(2) Indiquez le plugin du fournisseur de sauvegarde, par exemple **gcp**, s'il existe.

3. Le plugin **csi** est obligatoire pour sauvegarder les PV avec des instantanés CSI. Le plugin **csi** utilise les API d'instantanés Velero CSI beta. Il n'est pas nécessaire de configurer un
4. Le plugin **openshift** est obligatoire.
5. Définissez **false** si vous souhaitez désactiver l'installation de Restic. Restic déploie un ensemble de démons, ce qui signifie que chaque nœud de travailleur a des pods **Restic** en cours d'exécution. Vous configurez Restic pour les sauvegardes en ajoutant **spec.defaultVolumesToRestic: true** au CR **Backup**.
6. Spécifier le sélecteur de nœuds à fournir à Restic podSpec
7. Spécifiez le fournisseur de sauvegarde.
8. Si vous utilisez un plugin par défaut pour le fournisseur de sauvegarde, vous devez spécifier le nom par défaut correct pour **Secret**, par exemple, **cloud-credentials-gcp**. Si vous spécifiez un nom personnalisé, celui-ci est utilisé pour l'emplacement de sauvegarde. Si vous n'indiquez pas de nom **Secret**, le nom par défaut est utilisé.
9. Spécifiez un bac comme emplacement de stockage des sauvegardes. Si le bac n'est pas un bac dédié aux sauvegardes Velero, vous devez spécifier un préfixe.
10. Spécifiez un préfixe pour les sauvegardes Velero, par exemple, **velero**, si le seau est utilisé à des fins multiples.

4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s
```

```
NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                     1         1         1       96s
```

15.1.4.1. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

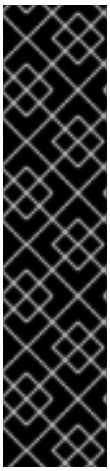
```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi 1
```

1 Ajouter le plugin par défaut **csi**.

15.1.5. Désinstallation de l'OADP

Vous désinstallez OpenShift API for Data Protection (OADP) en supprimant l'opérateur OADP. Pour plus d'informations, reportez-vous à la section Suppression des opérateurs d'un cluster .

15.2. SAUVEGARDE ET RESTAURATION DES MACHINES VIRTUELLES



IMPORTANT

OADP pour OpenShift Virtualization est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir des commentaires pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

Vous sauvegardez et restaurez les machines virtuelles en utilisant l'API OpenShift pour la protection des données (OADP).

Conditions préalables

- Accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

1. Installez l'opérateur OADP conformément aux instructions de votre fournisseur de services de stockage.
2. Installez l'application de protection des données avec les plugins **kubevirt** et **openshift**.
3. Sauvegarder les machines virtuelles en créant une ressource personnalisée (CR) sur **Backup** .
4. Rétablir le CR **Backup** en créant un CR**Restore** .

15.2.1. Ressources supplémentaires

- Fonctionnalités et plugins de l'OADP
- Dépannage

15.3. SAUVEGARDE DES MACHINES VIRTUELLES

IMPORTANT

OADP pour OpenShift Virtualization est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir des commentaires pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

Vous sauvegardez des machines virtuelles (VM) en créant une ressource personnalisée (CR) OpenShift API for Data Protection (OADP) **Backup**.

Le CR **Backup** effectue les actions suivantes :

- Sauvegarde les ressources d'OpenShift Virtualization en créant un fichier d'archive sur un stockage objet compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio.
- Sauvegarde des disques VM à l'aide de l'une des options suivantes :
  - Les instantanés de l'interface de stockage de conteneurs (CSI) sur le stockage en nuage compatible CSI, tel que Ceph RBD ou Ceph FS.
  - Sauvegardes de systèmes de fichiers résilients sur des systèmes de stockage d'objets.

NOTE

L'OADP fournit des crochets de sauvegarde pour geler le système de fichiers de la VM avant l'opération de sauvegarde et le dégeler lorsque la sauvegarde est terminée.

Le site **kubevirt-controller** crée les pods **virt-launcher** avec des annotations qui permettent à Velero d'exécuter le binaire **virt-freezer** avant et après l'opération de sauvegarde.

Les API **freeze** et **unfreeze** sont des sous-ressources de l'API VM snapshot. Pour plus de détails, voir À propos des instantanés de machines virtuelles .

Vous pouvez ajouter des crochets au CR **Backup** pour exécuter des commandes sur des machines virtuelles spécifiques avant ou après l'opération de sauvegarde.

Vous planifiez une sauvegarde en créant un CR**Schedule** au lieu d'un CR **Backup**.

15.3.1. Création d'un CR de sauvegarde

Vous sauvegardez les images Kubernetes, les images internes et les volumes persistants (PV) en créant une ressource personnalisée (CR) **Backup**.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.
- Conditions préalables relatives à l'emplacement de la sauvegarde :
  - Le stockage d'objets S3 doit être configuré pour Velero.
  - Un emplacement de sauvegarde doit être configuré dans le CR **DataProtectionApplication**.
- Conditions préalables pour l'emplacement des instantanés :
  - Votre fournisseur de cloud computing doit disposer d'une API d'instantané native ou prendre en charge les instantanés de l'interface de stockage de conteneurs (CSI).
  - Pour les instantanés CSI, vous devez créer un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.
  - Un emplacement de volume doit être configuré dans le CR **DataProtectionApplication**.

Procédure

1. Récupérez les CR **backupStorageLocations** en entrant la commande suivante :

```
$ oc get backupStorageLocations -n openshift-adp
```

Exemple de sortie

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. Créez un CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  hooks: {}
  includedNamespaces:
  - <namespace> 1
  includedResources: [] 2
  excludedResources: [] 3
  storageLocation: <velero-sample-1> 4
  ttl: 720h0m0s
  labelSelector: 5
  - matchLabels:
      app=<label_1>
  - matchLabels:
      app=<label_2>
  - matchLabels:
      app=<label_3>
  orlabelSelectors: 6
  - matchLabels:
      app=<label_1>
  - matchLabels:
      app=<label_2>
  - matchLabels:
      app=<label_3>
```

1. Spécifier un tableau d'espaces de noms à sauvegarder.
2. Facultatif : Spécifiez un tableau de ressources à inclure dans la sauvegarde. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement qualifiées. Si rien n'est spécifié, toutes les ressources sont incluses.
3. Facultatif : Spécifiez un tableau de ressources à exclure de la sauvegarde. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement qualifiées.
4. Indiquez le nom du CR **backupStorageLocations**.
5. Sauvegarde des ressources qui ont toutes les étiquettes spécifiées.
6. Sauvegarde des ressources qui ont une ou plusieurs des étiquettes spécifiées.

3. Vérifiez que l'état de la CR **Backup** est **Completed**:

```
$ oc get backup -n openshift-adp <backup> -o jsonpath='{.status.phase}'
```

15.3.1.1. Sauvegarde de volumes persistants avec des instantanés CSI

Vous sauvegardez des volumes persistants avec des instantanés de l'interface de stockage de conteneurs (CSI) en modifiant la ressource personnalisée (CR) **VolumeSnapshotClass** du stockage en nuage avant de créer la CR **Backup**.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.
- Vous devez activer le CSI sur le site **DataProtectionApplication** CR.

Procédure

- Ajouter la paire clé-valeur **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** à la CR **VolumeSnapshotClass**:

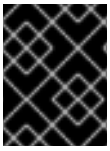
```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true"
driver: <csi_driver>
deletionPolicy: Retain
```

Vous pouvez maintenant créer un CR **Backup**.

15.3.1.2. Sauvegarde des applications avec Restic

Vous sauvegardez les ressources Kubernetes, les images internes et les volumes persistants avec Restic en modifiant la ressource personnalisée (CR) **Backup**.

Il n'est pas nécessaire de spécifier un emplacement d'instantané dans le CR **DataProtectionApplication**.



IMPORTANT

Restic ne prend pas en charge la sauvegarde des volumes **hostPath**. Pour plus d'informations, voir les limitations supplémentaires de Restic.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Vous ne devez pas désactiver l'installation par défaut de Restic en remplaçant **spec.configuration.restic.enable** par **false** dans le CR **DataProtectionApplication**.
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.

Procédure

- Modifiez le CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
```

```
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  defaultVolumesToRestic: true (1)
...
```

(1) Ajouter **defaultVolumesToRestic: true** au bloc **spec**.

15.3.1.3. Création de crochets de sauvegarde

Vous créez des crochets de sauvegarde pour exécuter des commandes dans un conteneur d'un pod en modifiant la ressource personnalisée (CR) **Backup**.

Pre s'exécutent avant la sauvegarde du pod. *Post* s'exécutent après la sauvegarde.

Procédure

- Ajoutez un crochet au bloc **spec.hooks** du CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> (1)
        excludedNamespaces: (2)
        - <namespace>
        includedResources: []
        - pods (3)
        excludedResources: [] (4)
        labelSelector: (5)
          matchLabels:
            app: velero
            component: server
        pre: (6)
          - exec:
              container: <container> (7)
              command:
              - /bin/uname (8)
              - -a
              onError: Fail (9)
              timeout: 30s (10)
        post: (11)
...
```

1. Facultatif : vous pouvez spécifier les espaces de noms auxquels le crochet s'applique. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.
2. Facultatif : vous pouvez spécifier des espaces de noms auxquels le crochet ne s'applique pas.
3. Actuellement, les pods sont la seule ressource prise en charge à laquelle les crochets peuvent s'appliquer.
4. Facultatif : vous pouvez spécifier les ressources auxquelles le crochet ne s'applique pas.
5. Facultatif : Ce crochet ne s'applique qu'aux objets correspondant à l'étiquette. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.
6. Tableau de crochets à exécuter avant la sauvegarde.
7. Facultatif : si le conteneur n'est pas spécifié, la commande s'exécute dans le premier conteneur du pod.
8. Il s'agit du point d'entrée du conteneur init ajouté.
9. Les valeurs autorisées pour le traitement des erreurs sont **Fail** et **Continue**. La valeur par défaut est **Fail**.
10. Facultatif : durée d'attente pour l'exécution des commandes. La valeur par défaut est **30s**.
11. Ce bloc définit un tableau de hooks à exécuter après la sauvegarde, avec les mêmes paramètres que les hooks de pré-sauvegarde.

15.3.2. Planification des sauvegardes

Vous planifiez les sauvegardes en créant une ressource personnalisée (CR) **Schedule** au lieu d'une CR **Backup**.

AVERTISSEMENT

Laissez suffisamment de temps dans votre calendrier de sauvegarde pour qu'une sauvegarde se termine avant qu'une autre ne soit créée.

Par exemple, si la sauvegarde d'un espace de noms prend généralement 10 minutes, ne planifiez pas de sauvegardes plus fréquentes que toutes les 15 minutes.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.

Procédure

1. Récupérer les CR de **backupStorageLocations**:

```
$ oc get backupStorageLocations -n openshift-adp
```

Exemple de sortie

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. Créez un CR **Schedule**, comme dans l'exemple suivant :

```
$ cat << EOF | oc apply -f -
apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: <schedule>
  namespace: openshift-adp
spec:
  schedule: 0 7 * * * 1
  template:
    hooks: {}
    includedNamespaces:
    - <namespace> 2
    storageLocation: <velero-sample-1> 3
    defaultVolumesToRestic: true 4
    ttl: 720h0m0s
EOF
```

1. **cron** expression to schedule the backup, for example, **0 7 * * *** to perform a backup every day at 7:00.
2. Tableau des espaces de noms à sauvegarder.
3. Nom du CR **backupStorageLocations**.
4. Facultatif : Ajoutez la paire clé-valeur **defaultVolumesToRestic: true** si vous sauvegardez des volumes avec Restic.

3. Vérifiez que l'état de **Schedule** CR est **Completed** après l'exécution de la sauvegarde programmée :

```
$ oc get schedule -n openshift-adp <schedule> -o jsonpath='{.status.phase}'
```

15.3.3. Ressources supplémentaires

- Vue d'ensemble des instantanés de volume CSI

15.4. RESTAURATION DES MACHINES VIRTUELLES

Vous restaurez une ressource personnalisée (CR) OpenShift API for Data Protection (OADP) **Backup** en créant une CR**Restore** .

Vous pouvez ajouter des crochets au CR **Restore** pour exécuter des commandes dans les conteneurs d'initialisation, avant le démarrage du conteneur d'application ou dans le conteneur d'application lui-même.

15.4.1. Création d'un CR de restauration

Vous restaurez une ressource personnalisée (CR) **Backup** en créant une CR **Restore**.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.
- Vous devez avoir un Velero **Backup** CR.
- Ajustez la taille demandée pour que la capacité du volume persistant (PV) corresponde à la taille demandée au moment de la sauvegarde.

Procédure

1. Créez un CR **Restore**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  backupName: <backup> 1
  includedResources: [] 2
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  restorePVs: true
```

1. Nom du CR **Backup**.
2. Facultatif. Spécifiez un tableau de ressources à inclure dans le processus de restauration. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement qualifiées. Si rien n'est spécifié, toutes les ressources sont incluses.

2. Vérifiez que l'état du CR **Restore** est **Completed** en entrant la commande suivante :

```
$ oc get restore -n openshift-adp <restore> -o jsonpath='{.status.phase}'
```

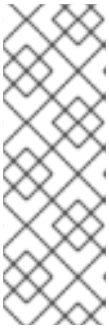
3. Vérifiez que les ressources de sauvegarde ont été restaurées en entrant la commande suivante :

```
$ oc get all -n <namespace> 1
```

1 Namespace que vous avez sauvegardé.

4. Si vous utilisez Restic pour restaurer les objets **DeploymentConfig** ou si vous utilisez des crochets post-restauration, exécutez le script de nettoyage **dc-restic-post-restore.sh** en entrant la commande suivante :

```
bash dc-restic-post-restore.sh <restore-name>
```



NOTE

Au cours du processus de restauration, les plug-ins OADP Velero réduisent les objets **DeploymentConfig** et restaurent les pods en tant que pods autonomes pour éviter que le cluster ne supprime les pods **DeploymentConfig** restaurés immédiatement après la restauration et pour permettre aux hooks Restic et post-restauration de terminer leurs actions sur les pods restaurés. Le script de nettoyage supprime ces pods déconnectés et met à l'échelle tous les objets **DeploymentConfig** jusqu'au nombre approprié de répliques.

Exemple 15.1. dc-restic-post-restore.sh script de nettoyage

```
#!/bin/bash
set -e

# if sha256sum exists, use it to check the integrity of the file
if command -v sha256sum >/dev/null 2>&1; then
  CHECKSUM_CMD="sha256sum"
else
  CHECKSUM_CMD="shasum -a 256"
fi

label_name () {
    if [ "${#1}" -le "63" ]; then
 echo $1
 return
    fi
    sha=$(echo -n $1|$CHECKSUM_CMD)
    echo "${1:0:57}${sha:0:6}"
}

OADP_NAMESPACE=${OADP_NAMESPACE:=openshift-adp}

if [[ $# -ne 1 ]]; then
    echo "usage: ${BASH_SOURCE} restore-name"
    exit 1
fi

echo using OADP Namespace $OADP_NAMESPACE
echo restore: $1

label=$(label_name $1)
echo label: $label

echo Deleting disconnected restore pods
```

```
oc delete pods -l oadp.openshift.io/disconnected-from-dc=$label

for dc in $(oc get dc --all-namespaces -l oadp.openshift.io/replicas-modified=$label -o
jsonpath='{range .items[*]}{.metadata.namespace}{","}{.metadata.name}{","}
{.metadata.annotations.oadp\.openshift\.io/original-replicas}{","}
{.metadata.annotations.oadp\.openshift\.io/original-paused}{"\n"}')
do
    IFS=',' read -ra dc_arr <<< "$dc"
    if [ ${#dc_arr[0]} -gt 0 ]; then
 echo Found deployment ${dc_arr[0]}/${dc_arr[1]}, setting replicas: ${dc_arr[2]}, paused:
${dc_arr[3]}
 cat <<EOF | oc patch dc  -n ${dc_arr[0]} ${dc_arr[1]} --patch-file /dev/stdin
spec:
  replicas: ${dc_arr[2]}
  paused: ${dc_arr[3]}
EOF
    fi
done
```

15.4.1.1. Création de crochets de restauration

Vous créez des crochets de restauration pour exécuter des commandes dans un conteneur dans un pod tout en restaurant votre application en modifiant la ressource personnalisée (CR) **Restore**.

Vous pouvez créer deux types de crochets de restauration :

- Un crochet **init** ajoute un conteneur init à un pod pour effectuer des tâches de configuration avant que le conteneur d'application ne démarre.
  Si vous restaurez une sauvegarde Restic, le conteneur init **restic-wait** est ajouté avant le conteneur init restore hook.

- Un hook **exec** exécute des commandes ou des scripts dans un conteneur d'un pod restauré.

Procédure

- Ajoutez un crochet au bloc **spec.hooks** du CR **Restore**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces:
        - <namespace>
        includedResources:
        - pods 2
        excludedResources: []
        labelSelector: 3
```

```
      matchLabels:
        app: velero
        component: server
    postHooks:
    - init:
        initContainers:
        - name: restore-hook-init
          image: alpine:latest
          volumeMounts:
          - mountPath: /restores/pvc1-vm
            name: pvc1-vm
          command:
          - /bin/ash
          - -c
        timeout: 4
    - exec:
        container: <container> 5
        command:
        - /bin/bash 6
        - -c
        - "psql < /backup/backup.sql"
        waitTimeout: 5m 7
        execTimeout: 1m 8
        onError: Continue 9
```

1. Facultatif : Tableau des espaces de noms auxquels le crochet s'applique. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.
2. Actuellement, les pods sont la seule ressource prise en charge à laquelle les crochets peuvent s'appliquer.
3. Facultatif : Ce crochet ne s'applique qu'aux objets correspondant au sélecteur d'étiquette.
4. Facultatif : Timeout indique la durée maximale pendant laquelle Velero attend la fin de **initContainers**.
5. Facultatif : si le conteneur n'est pas spécifié, la commande s'exécute dans le premier conteneur du pod.
6. Il s'agit du point d'entrée du conteneur init ajouté.
7. Facultatif : durée d'attente pour qu'un conteneur soit prêt. Cette durée doit être suffisante pour que le conteneur démarre et que tous les crochets précédents dans le même conteneur soient terminés. S'il n'est pas défini, le processus de restauration attend indéfiniment.
8. Facultatif : durée d'attente pour l'exécution des commandes. La valeur par défaut est **30s**.
9. Les valeurs autorisées pour le traitement des erreurs sont **Fail** et **Continue**:
    - **Continue**: Seuls les échecs de commande sont consignés.
    - **Fail**: Plus aucun crochet de restauration n'est exécuté dans aucun conteneur, dans aucun pod. Le statut du CR **Restore** sera **PartiallyFailed**.