



OpenShift Container Platform 4.12

Support des conteneurs Windows pour OpenShift

Guide des conteneurs Red Hat OpenShift pour Windows

OpenShift Container Platform 4.12 Support des conteneurs Windows pour OpenShift

Guide des conteneurs Red Hat OpenShift pour Windows

Notice légale

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Red Hat OpenShift for Windows Containers fournit un support intégré pour l'exécution de conteneurs Microsoft Windows Server sur OpenShift Container Platform. Ce guide fournit tous les détails.

Table des matières

CHAPITRE 1. APERÇU DE LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT	3
CHAPITRE 2. NOTES DE VERSION SUR LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT	4
2.1. À PROPOS DE LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT	4
2.2. OBTENIR DE L'AIDE	4
2.3. NOTES DE VERSION POUR RED HAT WINDOWS MACHINE CONFIG OPERATOR 7.0.1	4
2.4. NOTES DE VERSION POUR RED HAT WINDOWS MACHINE CONFIG OPERATOR 7.0.0	4
2.5. CONDITIONS PRÉALABLES POUR WINDOWS MACHINE CONFIG OPERATOR	6
2.6. LIMITES CONNUES	7
CHAPITRE 3. COMPRENDRE LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS	9
3.1. CONDITIONS PRÉALABLES POUR WINDOWS MACHINE CONFIG OPERATOR	9
3.2. GESTION DE LA CHARGE DE TRAVAIL WINDOWS	11
3.3. SERVICES DE NŒUDS WINDOWS	12
3.4. LIMITES CONNUES	13
CHAPITRE 4. PERMETTRE LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS	15
Conditions préalables	15
4.1. INSTALLATION DE L'OPÉRATEUR WINDOWS MACHINE CONFIG	15
4.2. CONFIGURATION D'UN SECRET POUR L'OPÉRATEUR DE CONFIGURATION DE LA MACHINE WINDOWS	18
4.3. RESSOURCES SUPPLÉMENTAIRES	19
CHAPITRE 5. CRÉATION D'OBJETS WINDOWS MACHINESSET	20
5.1. CRÉATION D'UN OBJET WINDOWS MACHINESSET SUR AWS	20
5.2. CRÉATION D'UN OBJET WINDOWS MACHINESSET SUR AZURE	25
5.3. CRÉATION D'UN OBJET WINDOWS MACHINESSET SUR VSPHERE	31
5.4. CRÉATION D'UN OBJET WINDOWS MACHINESSET SUR GCP	41
CHAPITRE 6. PLANIFICATION DES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS	48
Conditions préalables	48
6.1. PLACEMENT DES NACELLES DE FENÊTRES	48
6.2. CRÉATION D'UN OBJET RUNTIMECLASS POUR ENCAPSULER LES MÉCANISMES DE PLANIFICATION	49
6.3. EXEMPLE DE DÉPLOIEMENT D'UNE CHARGE DE TRAVAIL DE CONTENEUR WINDOWS	50
6.4. MISE À L'ÉCHELLE MANUELLE D'UN ENSEMBLE DE MACHINES DE CALCUL	51
CHAPITRE 7. MISES À NIVEAU DES NŒUDS WINDOWS	54
7.1. MISE À NIVEAU DE L'OPÉRATEUR DE CONFIGURATION DE LA MACHINE WINDOWS	54
CHAPITRE 8. UTILISATION D'INSTANCES WINDOWS BYOH (BRING-YOUR-OWN-HOST) EN TANT QUE NŒUDS	55
8.1. CONFIGURATION D'UNE INSTANCE WINDOWS BYOH	55
8.2. SUPPRESSION DES INSTANCES WINDOWS BYOH	56
CHAPITRE 9. SUPPRESSION DES NŒUDS WINDOWS	57
9.1. SUPPRESSION D'UNE MACHINE SPÉCIFIQUE	57
CHAPITRE 10. DÉSACTIVER LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS	58
10.1. DÉSINSTALLATION DE L'OPÉRATEUR WINDOWS MACHINE CONFIG	58
10.2. SUPPRESSION DE L'ESPACE DE NOMS WINDOWS MACHINE CONFIG OPERATOR	58

CHAPITRE 1. APERÇU DE LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT

La prise en charge de Windows Containers par Red Hat OpenShift est une fonctionnalité offrant la possibilité d'exécuter des nœuds de calcul Windows dans un cluster OpenShift Container Platform. Ceci est possible en utilisant Red Hat Windows Machine Config Operator (WMCO) pour installer et gérer les nœuds Windows. Avec un abonnement Red Hat, vous pouvez obtenir un support pour l'exécution de charges de travail Windows dans OpenShift Container Platform. Les instances Windows déployées par le WMCO sont configurées avec le moteur d'exécution de conteneur containerd. Pour plus d'informations, consultez les [notes de mise à jour](#).

Vous pouvez ajouter des nœuds Windows en créant un [ensemble de machines de calcul](#) ou en spécifiant des instances Window BYOH (Bring-Your-Own-Host) existantes par le biais d'une [carte de configuration](#).



NOTE

Les ensembles de machines de calcul ne sont pas pris en charge pour les clusters bare metal ou agnostiques vis-à-vis des fournisseurs.

Pour les charges de travail incluant à la fois Linux et Windows, OpenShift Container Platform vous permet de déployer des charges de travail Windows s'exécutant sur des conteneurs Windows Server tout en fournissant des charges de travail Linux traditionnelles hébergées sur Red Hat Enterprise Linux CoreOS (RHCOS) ou Red Hat Enterprise Linux (RHEL). Pour plus d'informations, voir [démarrer avec les charges de travail de conteneurs Windows](#).

Vous avez besoin du WMCO pour exécuter des charges de travail Windows dans votre cluster. Le WMCO orchestre le processus de déploiement et de gestion des charges de travail Windows sur un cluster. Pour plus d'informations, voir [comment activer les charges de travail Windows en conteneur](#).

Vous pouvez créer un objet Windows **MachineSet** pour créer des jeux de machines Windows d'infrastructure et des machines connexes afin de pouvoir déplacer les charges de travail Windows prises en charge vers les nouvelles machines Windows. Vous pouvez créer un objet Windows **MachineSet** sur plusieurs plateformes.

Vous pouvez [planifier des charges de travail Windows](#) sur des nœuds de calcul Windows.

Vous pouvez [effectuer des mises à niveau de l'opérateur Windows Machine Config](#) pour vous assurer que vos nœuds Windows disposent des dernières mises à jour.

Vous pouvez [supprimer un nœud Windows](#) en supprimant une machine spécifique.

Vous pouvez [utiliser les instances Windows BYOH \(Bring-Your-Own-Host\)](#) pour réutiliser les VMs Windows Server et les amener sur OpenShift Container Platform. Les instances Windows BYOH sont utiles aux utilisateurs qui cherchent à atténuer les perturbations majeures en cas de mise hors ligne d'un serveur Windows. Vous pouvez utiliser les instances Windows BYOH en tant que nœuds sur OpenShift Container Platform 4.8 et les versions ultérieures.

Vous pouvez [désactiver les charges de travail des conteneurs Windows](#) en procédant comme suit :

- Désinstallation de l'opérateur Windows Machine Config
- Suppression de l'espace de noms Windows Machine Config Operator

CHAPITRE 2. NOTES DE VERSION SUR LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT

2.1. À PROPOS DE LA PRISE EN CHARGE DE WINDOWS CONTAINERS PAR RED HAT OPENSIFT

La prise en charge de Windows Containers par Red Hat OpenShift permet d'exécuter des nœuds de calcul Windows dans un cluster OpenShift Container Platform. L'exécution des charges de travail Windows est possible en utilisant Red Hat Windows Machine Config Operator (WMCO) pour installer et gérer les nœuds Windows. Lorsque des nœuds Windows sont disponibles, vous pouvez exécuter des charges de travail de conteneurs Windows dans OpenShift Container Platform.

Les notes de version pour la prise en charge de Windows Containers par Red Hat OpenShift suivent le développement de WMCO, qui fournit toutes les capacités de charge de travail des conteneurs Windows dans OpenShift Container Platform.

2.2. OBTENIR DE L'AIDE

La prise en charge de Red Hat OpenShift pour Windows Containers est fournie et disponible en tant que composant optionnel et installable. La prise en charge de Red Hat OpenShift pour Windows Containers ne fait pas partie de l'abonnement à OpenShift Container Platform. Elle nécessite un abonnement Red Hat supplémentaire et est prise en charge conformément à l'[étendue de la couverture](#) et aux [accords de niveau de service](#).

Vous devez disposer de cet abonnement séparé pour bénéficier de la prise en charge de Red Hat OpenShift pour les conteneurs Windows. Sans cet abonnement Red Hat supplémentaire, le déploiement de charges de travail de conteneurs Windows dans des clusters de production n'est pas pris en charge. Vous pouvez demander de l'aide via le [portail client de Red Hat](#).

Pour plus d'informations, consultez le document Red Hat OpenShift Container Platform Life Cycle Policy pour la [prise en charge de Windows Containers par Red Hat OpenShift](#).

Si vous ne disposez pas de cet abonnement Red Hat supplémentaire, vous pouvez utiliser le Community Windows Machine Config Operator, une distribution qui ne bénéficie pas d'un support officiel.

2.3. NOTES DE VERSION POUR RED HAT WINDOWS MACHINE CONFIG OPERATOR 7.0.1

Cette version du WMCO fournit de nouvelles fonctionnalités et des corrections de bugs pour l'exécution de nœuds de calcul Windows dans un cluster OpenShift Container Platform. Les composants du WMCO 7.0.1 ont été publiés dans [RHBA-2023:0748](#).

2.3.1. Bug fixes

- Auparavant, WMCO 7.0.0 ne prenait pas en charge l'exécution dans un espace de noms autre que **openshift-windows-machine-operator**. Avec cette correction, vous pouvez exécuter WMCO dans un espace de noms personnalisé et mettre à niveau les clusters qui ont WMCO installé dans un espace de noms personnalisé. ([OCPBUGS-5065](#))

2.4. NOTES DE VERSION POUR RED HAT WINDOWS MACHINE CONFIG OPERATOR 7.0.0

Cette version du WMCO fournit de nouvelles fonctionnalités et des corrections de bugs pour l'exécution de nœuds de calcul Windows dans un cluster OpenShift Container Platform. Les composants du WMCO 7.0.0 ont été publiés dans [RHSA-2022:9096](#).

2.4.1. Nouvelles fonctionnalités et améliorations

2.4.1.1. Windows Instance Config Daemon (WICD)

Le Windows Instance Config Daemon (WICD) effectue désormais un grand nombre de tâches qui étaient auparavant exécutées par le Windows Machine Config Bootstrapper (WMCB). Le WICD est installé sur vos nœuds Windows. Les utilisateurs n'ont pas besoin d'interagir avec le WICD et ne devraient pas constater de différence dans le fonctionnement de WMCO.

2.4.1.2. Prise en charge des clusters fonctionnant sur Google Cloud Platform

Vous pouvez désormais exécuter des nœuds Windows Server 2022 sur un cluster installé sur Google Cloud Platform (GCP). Vous pouvez créer un objet Windows **MachineSet** sur GCP pour héberger des nœuds de calcul Windows Server 2022. Pour plus d'informations, voir [Création d'un objet Windows MachineSet sur vSphere](#).

2.4.2. Bug fixes

- Auparavant, le redémarrage du WMCO dans un cluster avec des nœuds Windows en cours d'exécution entraînait la suppression du point de terminaison de l'exportateur Windows. De ce fait, chaque nœud Windows ne pouvait pas rapporter de données métriques. Après cette correction, le point de terminaison est conservé lorsque le WMCO est redémarré. Par conséquent, les données de métriques sont rapportées correctement après le redémarrage de WMCO. ([BZ#2107261](#))
- Auparavant, le test visant à déterminer si le service antivirus Windows Defender est en cours d'exécution vérifiait de manière incorrecte tout processus dont le nom commençait par Windows Defender, quel que soit son état. Cela entraînait une erreur lors de la création d'exclusions de pare-feu pour containerd sur des instances sans Windows Defender installé. Cette correction vérifie désormais la présence du processus spécifique en cours d'exécution associé au service antivirus Windows Defender. Par conséquent, le WMCO peut configurer correctement les instances Windows en tant que nœuds, que Windows Defender soit installé ou non. ([OCPBUGS-3573](#))

2.4.3. Problèmes connus

Les limitations connues suivantes ont été annoncées après la publication précédente de WMCO :

- OpenShift Serverless, Horizontal Pod Autoscaling et Vertical Pod Autoscaling ne sont pas pris en charge sur les nœuds Windows.
- La prise en charge de Red Hat OpenShift pour les conteneurs Windows ne prend pas en charge l'ajout de nœuds Windows à un cluster par le biais d'un port trunk. La seule configuration réseau prise en charge pour l'ajout de nœuds Windows est un port d'accès qui transporte le trafic pour le VLAN.
- WMCO 7.0.0 ne prend pas en charge l'exécution dans un espace de noms autre que **openshift-windows-machine-operator**. Si vous utilisez un espace de noms personnalisé, il est recommandé de ne pas mettre à niveau WMCO 7.0.0, mais plutôt WMCO 7.0.1 lorsqu'il sera

disponible. Si votre OCM est configuré avec la stratégie d'approbation des mises à jour **Automatic**, vous devez la remplacer par **Manual** pour l'OCM 7.0.0. Consultez les [instructions d'installation](#) pour plus d'informations sur la modification de la stratégie d'approbation.

Ressources supplémentaires

Voir la liste complète des [limitations connues](#)

2.5. CONDITIONS PRÉALABLES POUR WINDOWS MACHINE CONFIG OPERATOR

Les informations suivantes détaillent les versions de plate-forme, les versions de Windows Server et les configurations de réseau prises en charge par l'opérateur Windows Machine Config. Consultez la documentation vSphere pour toute information concernant uniquement cette plateforme.

2.5.1. Plateformes et versions de Windows Server supportées par WMCO 7.0.0

Le tableau suivant énumère les [versions de Windows Server](#) prises en charge par WMCO 6.0.0, en fonction de la plate-forme applicable. Les versions de Windows Server qui ne figurent pas dans la liste ne sont pas prises en charge et toute tentative d'utilisation entraînera des erreurs. Pour éviter ces erreurs, n'utilisez que la version appropriée à votre plate-forme.

Plate-forme	Version de Windows Server prise en charge
Amazon Web Services (AWS)	Windows Server 2019, version 1809
Microsoft Azure	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809
VMware vSphere	Windows Server 2022, OS Build 20348.681 ou version ultérieure
Google Cloud Platform (GCP)	Windows Server 2022, OS Build 20348.681 ou version ultérieure
Métal nu ou fournisseur agnostique	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809

2.5.2. Réseaux pris en charge

La mise en réseau hybride avec OVN-Kubernetes est la seule configuration de mise en réseau prise en charge. Voir les ressources supplémentaires ci-dessous pour plus d'informations sur cette fonctionnalité. Les tableaux suivants indiquent le type de configuration réseau et les versions de Windows Server à utiliser en fonction de votre plateforme. Vous devez spécifier la configuration réseau lorsque vous installez le cluster. Sachez que le réseau OpenShift SDN est le réseau par défaut pour les clusters OpenShift Container Platform. Cependant, OpenShift SDN n'est pas pris en charge par WMCO.

Tableau 2.1. Soutien à la mise en réseau des plates-formes

Plate-forme	Réseaux pris en charge
Amazon Web Services (AWS)	Réseau hybride avec OVN-Kubernetes
Microsoft Azure	Réseau hybride avec OVN-Kubernetes
VMware vSphere	Réseau hybride avec OVN-Kubernetes avec un port VXLAN personnalisé
Google Cloud Platform (GCP)	Réseau hybride avec OVN-Kubernetes
Métal nu ou fournisseur agnostique	Réseau hybride avec OVN-Kubernetes

Tableau 2.2. Prise en charge des serveurs Windows hybrides OVN-Kubernetes

Réseau hybride avec OVN-Kubernetes	Version de Windows Server prise en charge
Port VXLAN par défaut	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809
Port VXLAN personnalisé	Windows Server 2022, OS Build 20348.681 ou version ultérieure

2.6. LIMITES CONNUES

Notez les limitations suivantes lorsque vous travaillez avec des nœuds Windows gérés par le WMCO (nœuds Windows) :

- Les fonctionnalités suivantes d'OpenShift Container Platform ne sont pas prises en charge sur les nœuds Windows :
 - Constructions d'images
 - OpenShift Pipelines
 - OpenShift Service Mesh
 - Surveillance OpenShift des projets définis par l'utilisateur
 - OpenShift Serverless
 - Mise à l'échelle horizontale du pod
 - Mise à l'échelle automatique des nacelles verticales
- Les fonctionnalités Red Hat suivantes ne sont pas prises en charge sur les nœuds Windows :
 - [Gestion des coûts Red Hat](#)

- [Red Hat OpenShift Local](#)
- Les nœuds Windows ne prennent pas en charge l'extraction d'images de conteneurs à partir de registres privés. Vous pouvez utiliser des images provenant de registres publics ou prélever les images.
- Les nœuds Windows ne prennent pas en charge les charges de travail créées à l'aide de configurations de déploiement. Vous pouvez utiliser un déploiement ou une autre méthode pour déployer les charges de travail.
- Les nœuds Windows ne sont pas pris en charge dans les grappes qui utilisent un proxy à l'échelle de la grappe. En effet, le WMCO n'est pas en mesure d'acheminer le trafic via la connexion proxy pour les charges de travail.
- Les nœuds Windows ne sont pas pris en charge dans les grappes qui se trouvent dans un environnement déconnecté.
- La prise en charge de Red Hat OpenShift pour les conteneurs Windows ne prend pas en charge l'ajout de nœuds Windows à un cluster par le biais d'un port trunk. La seule configuration réseau prise en charge pour l'ajout de nœuds Windows est un port d'accès qui transporte le trafic pour le VLAN.
- La prise en charge de Red Hat OpenShift pour Windows Containers ne prend en charge que les pilotes de stockage in-tree pour tous les fournisseurs de cloud.
- Kubernetes a identifié les [limitations](#) suivantes [des fonctionnalités des nœuds](#):
 - Les pages volumineuses ne sont pas prises en charge pour les conteneurs Windows.
 - Les conteneurs privilégiés ne sont pas pris en charge pour les conteneurs Windows.
- Kubernetes a identifié [plusieurs problèmes de compatibilité des API](#).

CHAPITRE 3. COMPRENDRE LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS

La prise en charge de Red Hat OpenShift pour les conteneurs Windows fournit une prise en charge intégrée pour l'exécution de conteneurs Microsoft Windows Server sur OpenShift Container Platform. Pour ceux qui administrent des environnements hétérogènes avec un mélange de charges de travail Linux et Windows, OpenShift Container Platform vous permet de déployer des charges de travail Windows exécutées sur des conteneurs Windows Server tout en fournissant des charges de travail Linux traditionnelles hébergées sur Red Hat Enterprise Linux CoreOS (RHCOS) ou Red Hat Enterprise Linux (RHEL).



NOTE

Le multi-tenant pour les clusters qui ont des nœuds Windows n'est pas pris en charge. L'utilisation hostile du multi-tenant introduit des problèmes de sécurité dans tous les environnements Kubernetes. Des fonctionnalités de sécurité supplémentaires, telles que les [politiques de sécurité des pods](#) ou un contrôle d'accès basé sur les rôles (RBAC) plus fin pour les nœuds, rendent les exploits plus difficiles. Toutefois, si vous choisissez d'exécuter des charges de travail multi-tenant hostiles, un hyperviseur est la seule option de sécurité que vous devriez utiliser. Le domaine de sécurité de Kubernetes englobe l'ensemble du cluster, et non un nœud individuel. Pour ces types de charges de travail multi-locataires hostiles, vous devez utiliser des clusters physiquement isolés.

Les conteneurs Windows Server permettent d'isoler les ressources à l'aide d'un noyau partagé, mais ne sont pas destinés à être utilisés dans des scénarios de multi-location hostile. Les scénarios qui impliquent une multi-location hostile doivent utiliser les conteneurs isolés Hyper-V pour isoler fortement les locataires.

3.1. CONDITIONS PRÉALABLES POUR WINDOWS MACHINE CONFIG OPERATOR

Les informations suivantes détaillent les versions de plate-forme, les versions de Windows Server et les configurations de réseau prises en charge par l'opérateur Windows Machine Config. Consultez la documentation vSphere pour toute information concernant uniquement cette plateforme.

3.1.1. Plateformes et versions de Windows Server supportées par WMCO 7.0.0

Le tableau suivant énumère les [versions de Windows Server](#) prises en charge par WMCO 6.0.0, en fonction de la plate-forme applicable. Les versions de Windows Server qui ne figurent pas dans la liste ne sont pas prises en charge et toute tentative d'utilisation entraînera des erreurs. Pour éviter ces erreurs, n'utilisez que la version appropriée à votre plate-forme.

Plate-forme	Version de Windows Server prise en charge
Amazon Web Services (AWS)	Windows Server 2019, version 1809
Microsoft Azure	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809

Plate-forme	Version de Windows Server prise en charge
VMware vSphere	Windows Server 2022, OS Build 20348.681 ou version ultérieure
Google Cloud Platform (GCP)	Windows Server 2022, OS Build 20348.681 ou version ultérieure
Métal nu ou fournisseur agnostique	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809

3.1.2. Réseaux pris en charge

La mise en réseau hybride avec OVN-Kubernetes est la seule configuration de mise en réseau prise en charge. Voir les ressources supplémentaires ci-dessous pour plus d'informations sur cette fonctionnalité. Les tableaux suivants indiquent le type de configuration réseau et les versions de Windows Server à utiliser en fonction de votre plateforme. Vous devez spécifier la configuration réseau lorsque vous installez le cluster. Sachez que le réseau OpenShift SDN est le réseau par défaut pour les clusters OpenShift Container Platform. Cependant, OpenShift SDN n'est pas pris en charge par WMCO.

Tableau 3.1. Soutien à la mise en réseau des plates-formes

Plate-forme	Réseaux pris en charge
Amazon Web Services (AWS)	Réseau hybride avec OVN-Kubernetes
Microsoft Azure	Réseau hybride avec OVN-Kubernetes
VMware vSphere	Réseau hybride avec OVN-Kubernetes avec un port VXLAN personnalisé
Google Cloud Platform (GCP)	Réseau hybride avec OVN-Kubernetes
Métal nu ou fournisseur agnostique	Réseau hybride avec OVN-Kubernetes

Tableau 3.2. Prise en charge des serveurs Windows hybrides OVN-Kubernetes

Réseau hybride avec OVN-Kubernetes	Version de Windows Server prise en charge
Port VXLAN par défaut	<ul style="list-style-type: none"> Windows Server 2022, OS Build 20348.681 ou version ultérieure Windows Server 2019, version 1809
Port VXLAN personnalisé	Windows Server 2022, OS Build 20348.681 ou version ultérieure

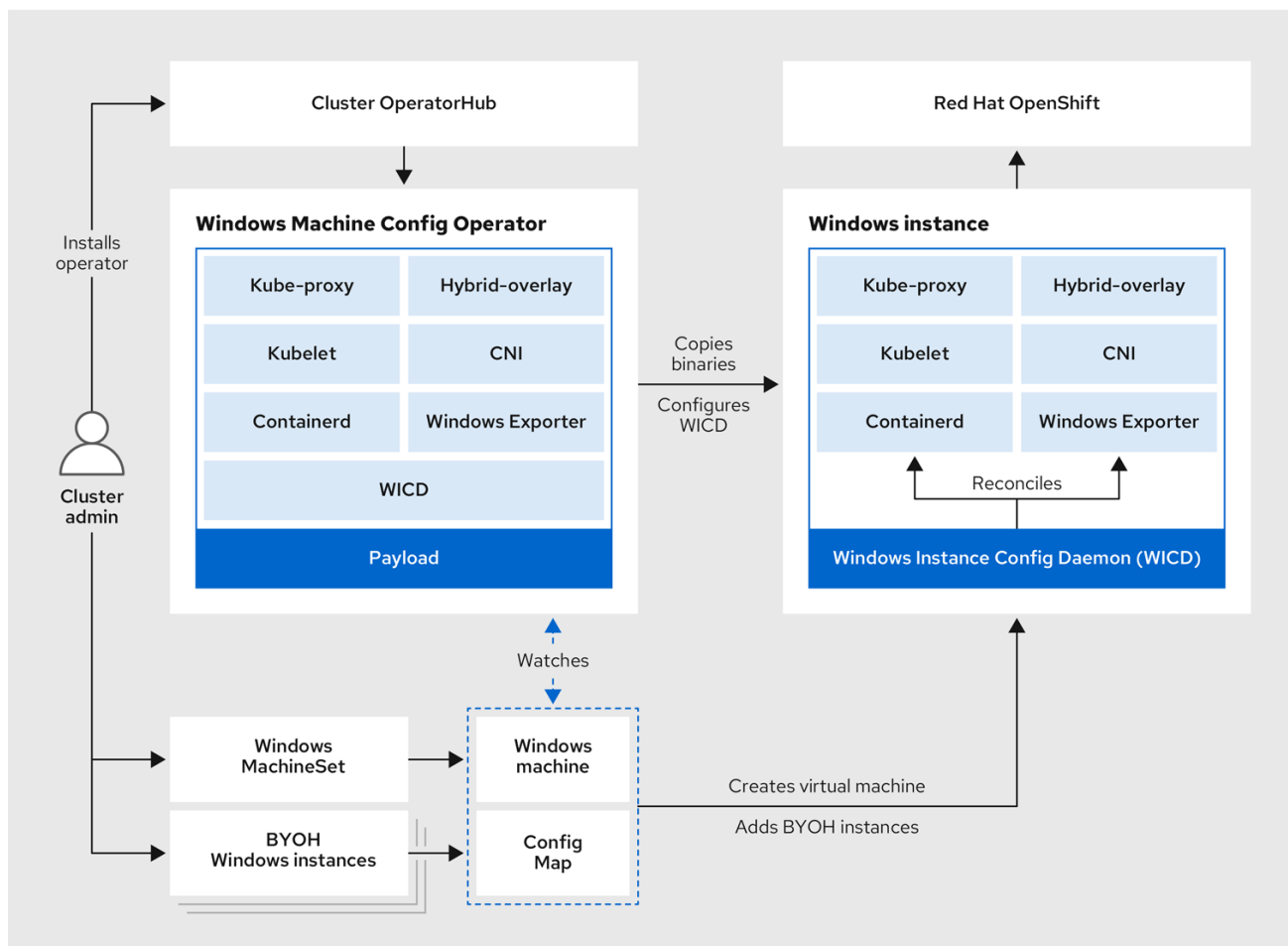
Ressources supplémentaires

- Voir [Configuration de la mise en réseau hybride avec OVN-Kubernetes](#)

3.2. GESTION DE LA CHARGE DE TRAVAIL WINDOWS

Pour exécuter des charges de travail Windows dans votre cluster, vous devez d'abord installer le Windows Machine Config Operator (WMCO). Le WMCO est un opérateur basé sur Linux qui s'exécute sur le plan de contrôle et les nœuds de calcul basés sur Linux. Il orchestre le processus de déploiement et de gestion des charges de travail Windows sur un cluster.

Figure 3.1. Conception WMCO

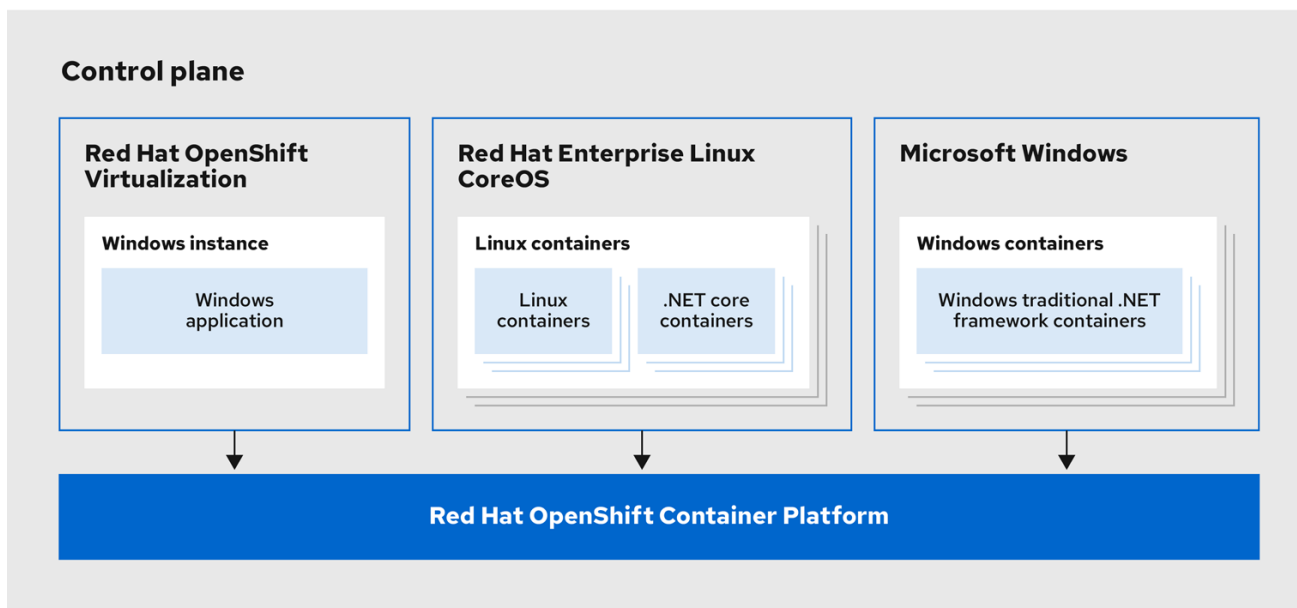


293_OpenShift_1122

Avant de déployer des charges de travail Windows, vous devez créer un nœud de calcul Windows et le faire rejoindre le cluster. Le nœud Windows héberge les charges de travail Windows dans un cluster et peut fonctionner avec d'autres nœuds de calcul basés sur Linux. Vous pouvez créer un nœud de calcul Windows en créant un ensemble de machines de calcul Windows pour héberger des machines de calcul Windows Server. Vous devez appliquer une étiquette spécifique à Windows à l'ensemble de machines de calcul qui spécifie une image de système d'exploitation Windows.

Le WMCO recherche les machines portant le label Windows. Une fois qu'un ensemble de machines de calcul Windows est détecté et que ses machines respectives sont approvisionnées, le WMCO configure la machine virtuelle (VM) Windows sous-jacente afin qu'elle puisse rejoindre le cluster en tant que nœud de calcul.

Figure 3.2. Charges de travail mixtes Windows et Linux



293_OpenShift_1122

L'OCMW attend un secret prédéterminé dans son espace de noms contenant une clé privée qui est utilisée pour interagir avec l'instance Windows. L'OCMW vérifie la présence de ce secret au moment du démarrage et crée un secret de données utilisateur que vous devez référencer dans l'objet Windows **MachineSet** que vous avez créé. Ensuite, l'OCMW remplit le secret de données utilisateur avec une clé publique qui correspond à la clé privée. Avec ces données en place, le cluster peut se connecter à la VM Windows en utilisant une connexion SSH.

Une fois que le cluster a établi une connexion avec la VM Windows, vous pouvez gérer le nœud Windows de la même manière qu'un nœud Linux.



NOTE

La console web d'OpenShift Container Platform offre la plupart des fonctionnalités de surveillance pour les nœuds Windows qui sont disponibles pour les nœuds Linux. Cependant, la possibilité de surveiller les graphiques de charge de travail pour les pods s'exécutant sur des nœuds Windows n'est pas disponible pour le moment.

L'ordonnancement des charges de travail Windows sur un nœud Windows peut se faire à l'aide des pratiques habituelles d'ordonnancement de pods telles que les taints, les tolérances et les sélecteurs de nœuds. Vous pouvez également différencier vos charges de travail Windows des charges de travail Linux et des autres charges de travail en version Windows en utilisant un objet **RuntimeClass**.

3.3. SERVICES DE NŒUDS WINDOWS

Les services suivants, spécifiques à Windows, sont installés sur chaque nœud Windows :

Service	Description
kubelet	Enregistre le nœud Windows et gère son état.

Service	Description
Plugins d'interface de réseau de conteneurs (CNI)	Expose la mise en réseau pour les nœuds Windows.
Windows Instance Config Daemon (WICD)	Maintient l'état de tous les services s'exécutant sur l'instance Windows afin de garantir que l'instance fonctionne comme un nœud de travail.
Windows Exporter	Exportation des métriques Prometheus à partir des nœuds Windows
Kubernetes Cloud Controller Manager (CCM)	Interagit avec la plateforme cloud Azure sous-jacente.
hybride-overlay	Crée le service de réseau hôte (HNS) de la plateforme OpenShift Container.
kube-proxy	Maintient les règles du réseau sur les nœuds permettant la communication avec l'extérieur.
containerd container runtime	Gère le cycle de vie complet des conteneurs.

3.4. LIMITES CONNUES

Notez les limitations suivantes lorsque vous travaillez avec des nœuds Windows gérés par le WMCO (nœuds Windows) :

- Les fonctionnalités suivantes d'OpenShift Container Platform ne sont pas prises en charge sur les nœuds Windows :
 - Constructions d'images
 - OpenShift Pipelines
 - OpenShift Service Mesh
 - Surveillance OpenShift des projets définis par l'utilisateur
 - OpenShift Serverless
 - Mise à l'échelle horizontale du pod
 - Mise à l'échelle automatique des nacelles verticales
- Les fonctionnalités Red Hat suivantes ne sont pas prises en charge sur les nœuds Windows :
 - [Gestion des coûts Red Hat](#)
 - [Red Hat OpenShift Local](#)
- Les nœuds Windows ne prennent pas en charge l'extraction d'images de conteneurs à partir de registres privés. Vous pouvez utiliser des images provenant de registres publics ou prélever les images.

- Les nœuds Windows ne prennent pas en charge les charges de travail créées à l'aide de configurations de déploiement. Vous pouvez utiliser un déploiement ou une autre méthode pour déployer les charges de travail.
- Les nœuds Windows ne sont pas pris en charge dans les grappes qui utilisent un proxy à l'échelle de la grappe. En effet, le WMCO n'est pas en mesure d'acheminer le trafic via la connexion proxy pour les charges de travail.
- Les nœuds Windows ne sont pas pris en charge dans les grappes qui se trouvent dans un environnement déconnecté.
- La prise en charge de Red Hat OpenShift pour les conteneurs Windows ne prend pas en charge l'ajout de nœuds Windows à un cluster par le biais d'un port trunk. La seule configuration réseau prise en charge pour l'ajout de nœuds Windows est un port d'accès qui transporte le trafic pour le VLAN.
- La prise en charge de Red Hat OpenShift pour Windows Containers ne prend en charge que les pilotes de stockage in-tree pour tous les fournisseurs de cloud.
- Kubernetes a identifié les [limitations](#) suivantes [des fonctionnalités des nœuds](#):
 - Les pages volumineuses ne sont pas prises en charge pour les conteneurs Windows.
 - Les conteneurs privilégiés ne sont pas pris en charge pour les conteneurs Windows.
- Kubernetes a identifié [plusieurs problèmes de compatibilité des API](#).

CHAPITRE 4. PERMETTRE LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS

Avant d'ajouter des charges de travail Windows à votre cluster, vous devez installer le Windows Machine Config Operator (WMCO), qui est disponible dans l'OperatorHub d'OpenShift Container Platform. Le WMCO orchestre le processus de déploiement et de gestion des charges de travail Windows sur un cluster.



NOTE

Le double NIC n'est pas supporté sur les instances Windows gérées par WMCO.

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 installé votre cluster à l'aide d'une infrastructure fournie par l'installateur ou à l'aide d'une infrastructure fournie par l'utilisateur avec le champ **platform: none** défini dans votre fichier **install-config.yaml**.
- Vous avez configuré la mise en réseau hybride avec OVN-Kubernetes pour votre cluster. Cette opération doit être réalisée lors de l'installation de votre cluster. Pour plus d'informations, voir [Configuration de la mise en réseau hybride](#).
- Vous exécutez un cluster OpenShift Container Platform version 4.6.8 ou ultérieure.



NOTE

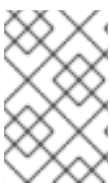
Les instances Windows déployées par le WMCO sont configurées avec le runtime containerd. Comme le WMCO installe et gère le runtime, il est recommandé de ne pas installer manuellement containerd sur les nœuds.

Ressources supplémentaires

- Pour connaître l'ensemble des conditions préalables à l'utilisation de Windows Machine Config Operator, voir [Comprendre les charges de travail des conteneurs Windows](#).

4.1. INSTALLATION DE L'OPÉRATEUR WINDOWS MACHINE CONFIG

Vous pouvez installer le Windows Machine Config Operator en utilisant la console web ou OpenShift CLI (**oc**).

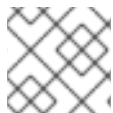


NOTE

Le WMCO n'est pas pris en charge dans les clusters qui utilisent un [proxy à l'échelle du cluster](#) car le WMCO n'est pas en mesure d'acheminer le trafic via la connexion proxy pour les charges de travail.

4.1.1. Installation de Windows Machine Config Operator à l'aide de la console web

Vous pouvez utiliser la console web d'OpenShift Container Platform pour installer le Windows Machine Config Operator (WMCO).

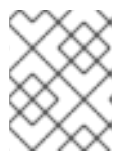


NOTE

Le double NIC n'est pas supporté sur les instances Windows gérées par WMCO.

Procédure

1. Depuis la perspective **Administrator** dans la console web de OpenShift Container Platform, naviguez jusqu'à la page **Operators → OperatorHub**.
2. Utilisez la boîte **Filter by keyword** pour rechercher **Windows Machine Config Operator** dans le catalogue. Cliquez sur la tuile **Windows Machine Config Operator**.
3. Examinez les informations relatives à l'opérateur et cliquez sur **Install**.
4. Sur la page **Install Operator**:
 - a. Sélectionnez le canal **stable** comme **Update Channel**. Le canal **stable** permet d'installer la dernière version stable du WMCO.
 - b. Le site **Installation Mode** est préconfiguré car le WMCO doit être disponible dans un seul espace de noms.
 - c. Choisissez l'espace de noms **Installed Namespace** pour le WMCO. L'espace de noms par défaut recommandé par l'opérateur est **openshift-windows-machine-config-operator**.
 - d. Cliquez sur la case à cocher **Enable Operator recommended cluster monitoring on the Namespace** pour activer la surveillance des clusters pour le WMCO.
 - e. Sélectionnez un site **Approval Strategy**.
 - La stratégie **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
 - La stratégie **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.
1. Cliquez sur **Install**. Le WMCO figure désormais sur la page **Installed Operators**.



NOTE

L'OCMW est installé automatiquement dans l'espace de noms que vous avez défini, comme **openshift-windows-machine-config-operator**.

2. Vérifiez que le site **Status** affiche **Succeeded** pour confirmer que l'installation de l'OCMW a réussi.

4.1.2. Installation de l'opérateur Windows Machine Config à l'aide du CLI

Vous pouvez utiliser la CLI OpenShift (**oc**) pour installer le Windows Machine Config Operator (WMCO).



NOTE

Le double NIC n'est pas supporté sur les instances Windows gérées par WMCO.

Procédure

1. Créer un espace de noms pour le WMCO.

a. Créer un fichier YAML de l'objet **Namespace** pour le WMCO. Par exemple, **wmco-namespace.yaml**:

```
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-windows-machine-config-operator 1
  labels:
    openshift.io/cluster-monitoring: "true" 2
```

1 Il est recommandé de déployer le WMCO dans l'espace de noms **openshift-windows-machine-config-operator**.

2 Cette étiquette est nécessaire pour activer la surveillance des clusters pour le WMCO.

b. Créer l'espace de noms :

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f wmco-namespace.yaml
```

2. Créez le groupe d'opérateurs pour le WMCO.

a. Créer un fichier YAML de l'objet **OperatorGroup**. Par exemple, **wmco-og.yaml**:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: windows-machine-config-operator
  namespace: openshift-windows-machine-config-operator
spec:
  targetNamespaces:
    - openshift-windows-machine-config-operator
```

b. Créez le groupe Opérateur :

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f wmco-og.yaml
```

3. Abonnement de l'espace de noms à l'OCMF.

- a. Créer un fichier YAML de l'objet **Subscription**. Par exemple, **wmco-sub.yaml**:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: windows-machine-config-operator
  namespace: openshift-windows-machine-config-operator
spec:
  channel: "stable" 1
  installPlanApproval: "Automatic" 2
  name: "windows-machine-config-operator"
  source: "redhat-operators" 3
  sourceNamespace: "openshift-marketplace" 4
```

- 1 Spécifiez **stable** comme canal.
- 2 Définissez une stratégie d'approbation. Vous pouvez définir **Automatic** ou **Manual**.
- 3 Spécifiez la source du catalogue **redhat-operators**, qui contient les manifestes de paquets **windows-machine-config-operator**. Si votre OpenShift Container Platform est installée sur un réseau restreint, également connu sous le nom de cluster déconnecté, indiquez le nom de l'objet **CatalogSource** que vous avez créé lors de la configuration de l'Operator Lifecycle Manager (OLM).
- 4 Espace de noms de la source de catalogue. Utilisez **openshift-marketplace** pour les sources de catalogue par défaut d'OperatorHub.

- b. Créer l'abonnement :

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f wmco-sub.yaml
```

Le WMCO est maintenant installé sur le site **openshift-windows-machine-config-operator**.

4. Vérifier l'installation de WMCO :

```
$ oc get csv -n openshift-windows-machine-config-operator
```

Exemple de sortie

```
NAME                                DISPLAY                                VERSION REPLACES PHASE
windows-machine-config-operator.2.0.0  Windows Machine Config Operator  2.0.0
Succeeded
```

4.2. CONFIGURATION D'UN SECRET POUR L'OPÉRATEUR DE CONFIGURATION DE LA MACHINE WINDOWS

Pour exécuter le Windows Machine Config Operator (WMCO), vous devez créer un secret dans l'espace de noms WMCO contenant une clé privée. Cette clé est nécessaire pour permettre à l'OCMW de communiquer avec la machine virtuelle Windows (VM).

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous avez créé un fichier codé PEM contenant une clé RSA.

Procédure

- Définir le secret requis pour accéder aux machines virtuelles Windows :

```
$ oc create secret generic cloud-private-key --from-file=private-  
key.pem=${HOME}/.ssh/<key> \  
-n openshift-windows-machine-config-operator 1
```

- 1 Vous devez créer la clé privée dans l'espace de noms WMCO, comme **openshift-windows-machine-config-operator**.

Il est recommandé d'utiliser une clé privée différente de celle utilisée lors de l'installation du cluster.

4.3. RESSOURCES SUPPLÉMENTAIRES

- [Génération d'une paire de clés pour l'accès SSH au nœud de cluster](#)
- [Ajout d'opérateurs à une grappe.](#)

CHAPITRE 5. CRÉATION D'OBJETS WINDOWS MACHINESSET

5.1. CRÉATION D'UN OBJET WINDOWS MACHINESSET SUR AWS

Vous pouvez créer un objet Windows **MachineSet** à des fins spécifiques dans votre cluster OpenShift Container Platform sur Amazon Web Services (AWS). Par exemple, vous pouvez créer des ensembles de machines Windows d'infrastructure et des machines connexes afin de pouvoir déplacer les charges de travail Windows de support vers les nouvelles machines Windows.

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous utilisez un serveur Windows pris en charge comme image du système d'exploitation. Utilisez la commande suivante **aws** pour interroger les images AMI valides :

```
$ aws ec2 describe-images --region <aws region name> --filters  
"Name=name,Values=Windows_Server-2019*English*Full*Containers*" "Name=is-  
public,Values=true" --query "reverse(sort_by(Images, &CreationDate))[*].{name : Name, id :  
ImageId}" --output table
```

5.1.1. Vue d'ensemble de l'API Machine

L'API Machine est une combinaison de ressources primaires basées sur le projet Cluster API en amont et de ressources OpenShift Container Platform personnalisées.

Pour les clusters OpenShift Container Platform 4.12, l'API Machine effectue toutes les actions de gestion du provisionnement de l'hôte du nœud une fois l'installation du cluster terminée. Grâce à ce système, OpenShift Container Platform 4.12 offre une méthode de provisionnement élastique et dynamique au-dessus d'une infrastructure de cloud public ou privé.

Les deux principales ressources sont les suivantes

Machines

Unité fondamentale décrivant l'hôte d'un nœud. Une machine possède une spécification **providerSpec**, qui décrit les types de nœuds de calcul proposés par les différentes plateformes de cloud computing. Par exemple, un type de machine pour un nœud de travail sur Amazon Web Services (AWS) peut définir un type de machine spécifique et les métadonnées requises.

Jeux de machines

MachineSet les ressources sont des groupes de machines de calcul. Les ensembles de machines de calcul sont aux machines de calcul ce que les ensembles de répliques sont aux pods. Si vous avez besoin de plus de machines de calcul ou si vous devez les réduire, vous modifiez le champ **replicas** de la ressource **MachineSet** pour répondre à vos besoins en matière de calcul.



AVERTISSEMENT

Les machines du plan de contrôle ne peuvent pas être gérées par des ensembles de machines de calcul.

Les ensembles de machines du plan de contrôle fournissent des capacités de gestion pour les machines du plan de contrôle prises en charge qui sont similaires à celles que les ensembles de machines de calcul fournissent pour les machines de calcul.

Pour plus d'informations, voir "Gestion des machines du plan de contrôle".

Les ressources personnalisées suivantes ajoutent des capacités supplémentaires à votre cluster :

Machine autoscaler

La ressource **MachineAutoscaler** met automatiquement à l'échelle les machines de calcul dans un nuage. Vous pouvez définir les limites minimales et maximales de mise à l'échelle pour les nœuds d'un ensemble de machines de calcul spécifié, et l'autoscaler de machines maintient cette plage de nœuds.

L'objet **MachineAutoscaler** prend effet après l'existence d'un objet **ClusterAutoscaler**. Les ressources **ClusterAutoscaler** et **MachineAutoscaler** sont mises à disposition par l'objet **ClusterAutoscalerOperator**.

Cluster autoscaler

Cette ressource est basée sur le projet de cluster autoscaler en amont. Dans l'implémentation d'OpenShift Container Platform, elle est intégrée à l'API Machine en étendant l'API compute machine set. Vous pouvez utiliser le cluster autoscaler pour gérer votre cluster de la manière suivante :

- Définir des limites d'échelle à l'échelle du cluster pour les ressources telles que les cœurs, les nœuds, la mémoire et les GPU
- Définir la priorité afin que le cluster donne la priorité aux pods et que de nouveaux nœuds ne soient pas mis en ligne pour des pods moins importants
- Définir la politique de mise à l'échelle de manière à pouvoir augmenter les nœuds mais pas les diminuer

Bilan de santé de la machine

La ressource **MachineHealthCheck** détecte lorsqu'une machine n'est pas saine, la supprime et, sur les plates-formes prises en charge, crée une nouvelle machine.

Dans OpenShift Container Platform version 3.11, il n'était pas possible de déployer facilement une architecture multizone car le cluster ne gérait pas le provisionnement des machines. À partir de la version 4.1 d'OpenShift Container Platform, ce processus est plus facile. Chaque ensemble de machines de calcul est limité à une seule zone, de sorte que le programme d'installation envoie des ensembles de machines de calcul à travers les zones de disponibilité en votre nom. Ainsi, comme votre calcul est dynamique, et en cas de défaillance d'une zone, vous disposez toujours d'une zone pour rééquilibrer vos

machines. Dans les régions Azure globales qui ne disposent pas de plusieurs zones de disponibilité, vous pouvez utiliser des ensembles de machines pour garantir une haute disponibilité. L'autoscaler assure l'équilibrage du meilleur effort pendant toute la durée de vie d'un cluster.

5.1.2. Exemple de YAML pour un objet Windows MachineSet sur AWS

Cet exemple YAML définit un objet Windows **MachineSet** fonctionnant sur Amazon Web Services (AWS) sur lequel le Windows Machine Config Operator (WMCO) peut réagir.

```

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <infrastructure_id>-windows-worker-<zone> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id> 3
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-windows-worker-<zone> 4
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id> 5
        machine.openshift.io/cluster-api-machine-role: worker
        machine.openshift.io/cluster-api-machine-type: worker
        machine.openshift.io/cluster-api-machineset: <infrastructure_id>-windows-worker-<zone> 6
        machine.openshift.io/os-id: Windows 7
    spec:
      metadata:
        labels:
          node-role.kubernetes.io/worker: "" 8
      providerSpec:
        value:
          ami:
            id: <windows_container_ami> 9
          apiVersion: awsproviderconfig.openshift.io/v1beta1
          blockDevices:
            - ebs:
                iops: 0
                volumeSize: 120
                volumeType: gp2
          credentialsSecret:
            name: aws-cloud-credentials
          deviceIndex: 0
          iamInstanceProfile:
            id: <infrastructure_id>-worker-profile 10
          instanceType: m5a.large
          kind: AWSMachineProviderConfig
          placement:
            availabilityZone: <zone> 11
            region: <region> 12

```

```

securityGroups:
  - filters:
    - name: tag:Name
      values:
        - <infrastructure_id>-worker-sg 13
  subnet:
    filters:
      - name: tag:Name
        values:
          - <infrastructure_id>-private-<zone> 14
    tags:
      - name: kubernetes.io/cluster/<infrastructure_id> 15
        value: owned
  userDataSecret:
    name: windows-user-data 16
    namespace: openshift-machine-api

```

1 3 5 10 13 14 15 Spécifiez l'ID de l'infrastructure qui est basé sur l'ID du cluster que vous avez défini lorsque vous avez approvisionné le cluster. Vous pouvez obtenir l'ID d'infrastructure en exécutant la commande suivante :

```
$ oc get -o jsonpath='{.status.infrastructureName}' infrastructure cluster
```

2 4 6 Spécifiez l'ID de l'infrastructure, l'étiquette du travailleur et la zone.

7 Configurez l'ensemble de machines de calcul comme une machine Windows.

8 Configurez le nœud Windows en tant que machine de calcul.

9 Indiquez l'identifiant AMI d'une image Windows prise en charge dans laquelle un moteur d'exécution de conteneur est installé.

11 Spécifiez la zone AWS, comme **us-east-1a**.

12 Spécifiez la région AWS, comme **us-east-1**.

16 Créé par le WMCO lorsqu'il configure la première machine Windows. Ensuite, le site **windows-user-data** est disponible pour tous les ensembles de machines de calcul suivants.

5.1.3. Création d'un ensemble de machines de calcul

En plus des ensembles de machines de calcul créés par le programme d'installation, vous pouvez créer vos propres ensembles pour gérer dynamiquement les ressources de calcul des machines pour les charges de travail spécifiques de votre choix.

Conditions préalables

- Déployer un cluster OpenShift Container Platform.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Créez un nouveau fichier YAML contenant l'échantillon de ressources personnalisées (CR) de l'ensemble de machines de calcul et nommé **<file_name>.yaml**.
Veillez à définir les valeurs des paramètres **<clusterID>** et **<role>**.
2. Facultatif : si vous n'êtes pas sûr de la valeur à définir pour un champ spécifique, vous pouvez vérifier un ensemble de machines de calcul existant dans votre cluster.
 - a. Pour répertorier les ensembles de machines de calcul de votre cluster, exécutez la commande suivante :

```
$ oc get machinesets -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0			55m
agl030519-vplxk-worker-us-east-1e	0	0			55m
agl030519-vplxk-worker-us-east-1f	0	0			55m

- b. Pour afficher les valeurs d'une ressource personnalisée (CR) d'un ensemble de machines de calcul spécifique, exécutez la commande suivante :

```
$ oc get machineset <machineset_name> \
-n openshift-machine-api -o yaml
```

Exemple de sortie

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
    name: <infrastructure_id>-<role> 2
    namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id>
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id>
        machine.openshift.io/cluster-api-machine-role: <role>
        machine.openshift.io/cluster-api-machine-type: <role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
    spec:
      providerSpec: 3
      ...
```

- 1 L'ID de l'infrastructure du cluster.
- 2 Une étiquette de nœud par défaut.



NOTE

Pour les clusters disposant d'une infrastructure fournie par l'utilisateur, un ensemble de machines de calcul ne peut créer que des machines de type **worker** et **infra**.

- 3 Les valeurs de la section **<providerSpec>** du CR de l'ensemble de machines de calcul sont spécifiques à la plate-forme. Pour plus d'informations sur les paramètres **<providerSpec>** dans le CR, consultez l'exemple de configuration du CR de l'ensemble de machines de calcul pour votre fournisseur.

3. Créez un CR **MachineSet** en exécutant la commande suivante :

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

Vérification

- Affichez la liste des ensembles de machines de calcul en exécutant la commande suivante :

```
$ oc get machineset -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-windows-worker-us-east-1a	1	1	1	1	11m
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0		55m	
agl030519-vplxk-worker-us-east-1e	0	0		55m	
agl030519-vplxk-worker-us-east-1f	0	0		55m	

Lorsque le nouveau jeu de machines de calcul est disponible, les valeurs **DESIRED** et **CURRENT** correspondent. Si le jeu de machines de calcul n'est pas disponible, attendez quelques minutes et exécutez à nouveau la commande.

5.1.4. Ressources supplémentaires

- Pour plus d'informations sur la gestion des jeux de machines, voir la section *Machine management*.

5.2. CRÉATION D'UN OBJET WINDOWSMACHINESET SUR AZURE

Vous pouvez créer un objet Windows **MachineSet** à des fins spécifiques dans votre cluster OpenShift Container Platform sur Microsoft Azure. Par exemple, vous pouvez créer des ensembles de machines Windows d'infrastructure et des machines connexes afin de pouvoir déplacer les charges de travail Windows de support vers les nouvelles machines Windows.

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous utilisez un serveur Windows pris en charge comme image du système d'exploitation.

5.2.1. Vue d'ensemble de l'API Machine

L'API Machine est une combinaison de ressources primaires basées sur le projet Cluster API en amont et de ressources OpenShift Container Platform personnalisées.

Pour les clusters OpenShift Container Platform 4.12, l'API Machine effectue toutes les actions de gestion du provisionnement de l'hôte du nœud une fois l'installation du cluster terminée. Grâce à ce système, OpenShift Container Platform 4.12 offre une méthode de provisionnement élastique et dynamique au-dessus d'une infrastructure de cloud public ou privé.

Les deux principales ressources sont les suivantes

Machines

Unité fondamentale décrivant l'hôte d'un nœud. Une machine possède une spécification **providerSpec**, qui décrit les types de nœuds de calcul proposés par les différentes plateformes de cloud computing. Par exemple, un type de machine pour un nœud de travail sur Amazon Web Services (AWS) peut définir un type de machine spécifique et les métadonnées requises.

Jeux de machines

MachineSet les ressources sont des groupes de machines de calcul. Les ensembles de machines de calcul sont aux machines de calcul ce que les ensembles de répliques sont aux pods. Si vous avez besoin de plus de machines de calcul ou si vous devez les réduire, vous modifiez le champ **replicas** de la ressource **MachineSet** pour répondre à vos besoins en matière de calcul.



AVERTISSEMENT

Les machines du plan de contrôle ne peuvent pas être gérées par des ensembles de machines de calcul.

Les ensembles de machines du plan de contrôle fournissent des capacités de gestion pour les machines du plan de contrôle prises en charge qui sont similaires à celles que les ensembles de machines de calcul fournissent pour les machines de calcul.

Pour plus d'informations, voir "Gestion des machines du plan de contrôle".

Les ressources personnalisées suivantes ajoutent des capacités supplémentaires à votre cluster :

Machine autoscaler

La ressource **MachineAutoscaler** met automatiquement à l'échelle les machines de calcul dans un nuage. Vous pouvez définir les limites minimales et maximales de mise à l'échelle pour les nœuds d'un ensemble de machines de calcul spécifié, et l'autoscaler de machines maintient cette plage de nœuds.

L'objet **MachineAutoscaler** prend effet après l'existence d'un objet **ClusterAutoscaler**. Les ressources **ClusterAutoscaler** et **MachineAutoscaler** sont mises à disposition par l'objet **ClusterAutoscalerOperator**.

Cluster autoscaler

Cette ressource est basée sur le projet de cluster autoscaler en amont. Dans l'implémentation d'OpenShift Container Platform, elle est intégrée à l'API Machine en étendant l'API compute machine set. Vous pouvez utiliser le cluster autoscaler pour gérer votre cluster de la manière suivante :

- Définir des limites d'échelle à l'échelle du cluster pour les ressources telles que les cœurs, les nœuds, la mémoire et les GPU
- Définir la priorité afin que le cluster donne la priorité aux pods et que de nouveaux nœuds ne soient pas mis en ligne pour des pods moins importants
- Définir la politique de mise à l'échelle de manière à pouvoir augmenter les nœuds mais pas les diminuer

Bilan de santé de la machine

La ressource **MachineHealthCheck** détecte lorsqu'une machine n'est pas saine, la supprime et, sur les plates-formes prises en charge, crée une nouvelle machine.

Dans OpenShift Container Platform version 3.11, il n'était pas possible de déployer facilement une architecture multizone car le cluster ne gérait pas le provisionnement des machines. À partir de la version 4.1 d'OpenShift Container Platform, ce processus est plus facile. Chaque ensemble de machines de calcul est limité à une seule zone, de sorte que le programme d'installation envoie des ensembles de machines de calcul à travers les zones de disponibilité en votre nom. Ainsi, comme votre calcul est dynamique, et en cas de défaillance d'une zone, vous disposez toujours d'une zone pour rééquilibrer vos machines. Dans les régions Azure globales qui ne disposent pas de plusieurs zones de disponibilité, vous pouvez utiliser des ensembles de machines pour garantir une haute disponibilité. L'autoscaler assure l'équilibrage du meilleur effort pendant toute la durée de vie d'un cluster.

5.2.2. Exemple de YAML pour un objet Windows MachineSet sur Azure

Cet exemple YAML définit un objet Windows **MachineSet** fonctionnant sur Microsoft Azure et sur lequel le Windows Machine Config Operator (WMCO) peut réagir.

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <windows_machine_set_name> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id> 3
      machine.openshift.io/cluster-api-machineset: <windows_machine_set_name> 4
  template:
    metadata:
      labels:
```

```

machine.openshift.io/cluster-api-cluster: <infrastructure_id> 5
machine.openshift.io/cluster-api-machine-role: worker
machine.openshift.io/cluster-api-machine-type: worker
machine.openshift.io/cluster-api-machineset: <windows_machine_set_name> 6
machine.openshift.io/os-id: Windows 7
spec:
  metadata:
    labels:
      node-role.kubernetes.io/worker: "" 8
  providerSpec:
    value:
      apiVersion: azureproviderconfig.openshift.io/v1beta1
      credentialsSecret:
        name: azure-cloud-credentials
        namespace: openshift-machine-api
      image: 9
        offer: WindowsServer
        publisher: MicrosoftWindowsServer
        resourceID: ""
        sku: 2019-Datacenter-with-Containers
        version: latest
      kind: AzureMachineProviderSpec
      location: <location> 10
      managedIdentity: <infrastructure_id>-identity 11
      networkResourceGroup: <infrastructure_id>-rg 12
      osDisk:
        diskSizeGB: 128
        managedDisk:
          storageAccountType: Premium_LRS
        osType: Windows
      publicIP: false
      resourceGroup: <infrastructure_id>-rg 13
      subnet: <infrastructure_id>-worker-subnet
      userDataSecret:
        name: windows-user-data 14
        namespace: openshift-machine-api
      vmSize: Standard_D2s_v3
      vnet: <infrastructure_id>-vnet 15
      zone: "<zone>" 16

```

- 1 3 5 11 12 13 15 Spécifiez l'ID de l'infrastructure qui est basé sur l'ID du cluster que vous avez défini lorsque vous avez approvisionné le cluster. Vous pouvez obtenir l'ID d'infrastructure en exécutant la commande suivante :

```
$ oc get -o jsonpath='{.status.infrastructureName}' infrastructure cluster
```

- 2 4 6 Indiquez le nom de l'ensemble de machines de calcul Windows. Les noms de machines Windows sur Azure ne peuvent pas comporter plus de 15 caractères. Par conséquent, le nom de l'ensemble de machines de calcul ne peut pas dépasser 9 caractères, en raison de la façon dont les noms de machines sont générés à partir de ce nom.
- 7 Configurez l'ensemble de machines de calcul comme une machine Windows.
- 8 Configurez le nœud Windows en tant que machine de calcul.

- 9 Spécifiez une offre d'image **WindowsServer** qui définit l'UGS **2019-Datacenter-with-Containers**.
- 10 Spécifiez la région Azure, comme **centralus**.
- 14 Créé par le WMCO lorsqu'il configure la première machine Windows. Ensuite, le site **windows-user-data** est disponible pour tous les ensembles de machines de calcul suivants.
- 16 Indiquez la zone de votre région où placer les machines. Assurez-vous que votre région prend en charge la zone que vous spécifiez.

5.2.3. Création d'un ensemble de machines de calcul

En plus des ensembles de machines de calcul créés par le programme d'installation, vous pouvez créer vos propres ensembles pour gérer dynamiquement les ressources de calcul des machines pour les charges de travail spécifiques de votre choix.

Conditions préalables

- Déployer un cluster OpenShift Container Platform.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Créez un nouveau fichier YAML contenant l'échantillon de ressources personnalisées (CR) de l'ensemble de machines de calcul et nommé **<file_name>.yaml**.
Veillez à définir les valeurs des paramètres **<clusterID>** et **<role>**.
2. Facultatif : si vous n'êtes pas sûr de la valeur à définir pour un champ spécifique, vous pouvez vérifier un ensemble de machines de calcul existant dans votre cluster.
 - a. Pour répertorier les ensembles de machines de calcul de votre cluster, exécutez la commande suivante :

```
$ oc get machinesets -n openshift-machine-api
```

Exemple de sortie

```
NAME                                DESIRED  CURRENT  READY  AVAILABLE  AGE
agl030519-vplxk-worker-us-east-1a  1        1        1      1          55m
agl030519-vplxk-worker-us-east-1b  1        1        1      1          55m
agl030519-vplxk-worker-us-east-1c  1        1        1      1          55m
agl030519-vplxk-worker-us-east-1d  0        0                55m
agl030519-vplxk-worker-us-east-1e  0        0                55m
agl030519-vplxk-worker-us-east-1f  0        0                55m
```

- b. Pour afficher les valeurs d'une ressource personnalisée (CR) d'un ensemble de machines de calcul spécifique, exécutez la commande suivante :

```
$ oc get machineset <machineset_name> \
-n openshift-machine-api -o yaml
```

Exemple de sortie

```

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> ❶
    name: <infrastructure_id>-<role> ❷
    namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id>
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id>
        machine.openshift.io/cluster-api-machine-role: <role>
        machine.openshift.io/cluster-api-machine-type: <role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
    spec:
      providerSpec: ❸
      ...

```

- ❶ L'ID de l'infrastructure du cluster.
- ❷ Une étiquette de nœud par défaut.



NOTE

Pour les clusters disposant d'une infrastructure fournie par l'utilisateur, un ensemble de machines de calcul ne peut créer que des machines de type **worker** et **infra**.

- ❸ Les valeurs de la section **<providerSpec>** du CR de l'ensemble de machines de calcul sont spécifiques à la plate-forme. Pour plus d'informations sur les paramètres **<providerSpec>** dans le CR, consultez l'exemple de configuration du CR de l'ensemble de machines de calcul pour votre fournisseur.

3. Créez un CR **MachineSet** en exécutant la commande suivante :

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

Vérification

- Affichez la liste des ensembles de machines de calcul en exécutant la commande suivante :

```
$ oc get machineset -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-windows-worker-us-east-1a	1	1	1	1	11m
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0			55m
agl030519-vplxk-worker-us-east-1e	0	0			55m
agl030519-vplxk-worker-us-east-1f	0	0			55m

Lorsque le nouveau jeu de machines de calcul est disponible, les valeurs **DESIRED** et **CURRENT** correspondent. Si le jeu de machines de calcul n'est pas disponible, attendez quelques minutes et exécutez à nouveau la commande.

5.2.4. Ressources supplémentaires

- Pour plus d'informations sur la gestion des jeux de machines, voir la section *Machine management*.

5.3. CRÉATION D'UN OBJET WINDOWS MACHINESET SUR VSPHERE

Vous pouvez créer un objet Windows **MachineSet** à des fins spécifiques dans votre cluster OpenShift Container Platform sur VMware vSphere. Par exemple, vous pouvez créer des ensembles de machines Windows d'infrastructure et des machines connexes afin de pouvoir déplacer les charges de travail Windows de support vers les nouvelles machines Windows.

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous utilisez un serveur Windows pris en charge comme image du système d'exploitation.

5.3.1. Vue d'ensemble de l'API Machine

L'API Machine est une combinaison de ressources primaires basées sur le projet Cluster API en amont et de ressources OpenShift Container Platform personnalisées.

Pour les clusters OpenShift Container Platform 4.12, l'API Machine effectue toutes les actions de gestion du provisionnement de l'hôte du nœud une fois l'installation du cluster terminée. Grâce à ce système, OpenShift Container Platform 4.12 offre une méthode de provisionnement élastique et dynamique au-dessus d'une infrastructure de cloud public ou privé.

Les deux principales ressources sont les suivantes

Machines

Unité fondamentale décrivant l'hôte d'un nœud. Une machine possède une spécification **providerSpec**, qui décrit les types de nœuds de calcul proposés par les différentes plateformes de cloud computing. Par exemple, un type de machine pour un nœud de travail sur Amazon Web Services (AWS) peut définir un type de machine spécifique et les métadonnées requises.

Jeux de machines

MachineSet les ressources sont des groupes de machines de calcul. Les ensembles de machines de calcul sont aux machines de calcul ce que les ensembles de répliques sont aux pods. Si vous avez besoin de plus de machines de calcul ou si vous devez les réduire, vous modifiez le champ **replicas** de la ressource **MachineSet** pour répondre à vos besoins en matière de calcul.



AVERTISSEMENT

Les machines du plan de contrôle ne peuvent pas être gérées par des ensembles de machines de calcul.

Les ensembles de machines du plan de contrôle fournissent des capacités de gestion pour les machines du plan de contrôle prises en charge qui sont similaires à celles que les ensembles de machines de calcul fournissent pour les machines de calcul.

Pour plus d'informations, voir "Gestion des machines du plan de contrôle".

Les ressources personnalisées suivantes ajoutent des capacités supplémentaires à votre cluster :

Machine autoscaler

La ressource **MachineAutoscaler** met automatiquement à l'échelle les machines de calcul dans un nuage. Vous pouvez définir les limites minimales et maximales de mise à l'échelle pour les nœuds d'un ensemble de machines de calcul spécifié, et l'autoscaler de machines maintient cette plage de nœuds.

L'objet **MachineAutoscaler** prend effet après l'existence d'un objet **ClusterAutoscaler**. Les ressources **ClusterAutoscaler** et **MachineAutoscaler** sont mises à disposition par l'objet **ClusterAutoscalerOperator**.

Cluster autoscaler

Cette ressource est basée sur le projet de cluster autoscaler en amont. Dans l'implémentation d'OpenShift Container Platform, elle est intégrée à l'API Machine en étendant l'API compute machine set. Vous pouvez utiliser le cluster autoscaler pour gérer votre cluster de la manière suivante :

- Définir des limites d'échelle à l'échelle du cluster pour les ressources telles que les cœurs, les nœuds, la mémoire et les GPU
- Définir la priorité afin que le cluster donne la priorité aux pods et que de nouveaux nœuds ne soient pas mis en ligne pour des pods moins importants
- Définir la politique de mise à l'échelle de manière à pouvoir augmenter les nœuds mais pas les diminuer

Bilan de santé de la machine

La ressource **MachineHealthCheck** détecte lorsqu'une machine n'est pas saine, la supprime et, sur les plates-formes prises en charge, crée une nouvelle machine.

Dans OpenShift Container Platform version 3.11, il n'était pas possible de déployer facilement une architecture multizone car le cluster ne gérait pas le provisionnement des machines. À partir de la version 4.1 d'OpenShift Container Platform, ce processus est plus facile. Chaque ensemble de machines de calcul est limité à une seule zone, de sorte que le programme d'installation envoie des ensembles de machines de calcul à travers les zones de disponibilité en votre nom. Ainsi, comme votre calcul est dynamique, et en cas de défaillance d'une zone, vous disposez toujours d'une zone pour rééquilibrer vos

machines. Dans les régions Azure globales qui ne disposent pas de plusieurs zones de disponibilité, vous pouvez utiliser des ensembles de machines pour garantir une haute disponibilité. L'autoscaler assure l'équilibrage du meilleur effort pendant toute la durée de vie d'un cluster.

5.3.2. Préparer votre environnement vSphere pour les charges de travail des conteneurs Windows

Vous devez préparer votre environnement vSphere pour les charges de travail des conteneurs Windows en créant l'image dorée vSphere Windows VM et en activant la communication avec le serveur API interne pour le WMCO.

5.3.2.1. Création de l'image dorée de la VM Windows vSphere

Créez une image dorée de machine virtuelle (VM) Windows vSphere.

Conditions préalables

- Vous avez créé une paire de clés privée/publique, qui est utilisée pour configurer l'authentification basée sur les clés dans le serveur OpenSSH. La clé privée doit également être configurée dans l'espace de noms Windows Machine Config Operator (WMCO). Cela est nécessaire pour permettre à l'opérateur de configuration de la machine Windows de communiquer avec la VM Windows. Voir la section "Configuration d'un secret pour l'opérateur de configuration de la machine Windows" pour plus de détails.



NOTE

Vous devez utiliser les commandes [Microsoft PowerShell](#) dans plusieurs cas lors de la création de votre VM Windows. Dans ce guide, les commandes PowerShell se distinguent par le préfixe **PS C:\>**.

Procédure

1. Sélectionnez une version de Windows Server compatible. Actuellement, la version stable de Windows Machine Config Operator (WMCO) prend en charge Windows Server 2022 Long-Term Servicing Channel avec le correctif [KB5012637](#) pour la mise en réseau des conteneurs au niveau du système d'exploitation.
2. Créez une nouvelle VM dans le client vSphere en utilisant l'image dorée de la VM avec une version compatible de Windows Server. Pour plus d'informations sur les versions compatibles, consultez la section "Windows Machine Config Operator prerequisites" des notes de mise à jour "Red Hat OpenShift support for Windows Containers release notes"



IMPORTANT

La version du matériel virtuel de votre VM doit répondre aux exigences de l'infrastructure d'OpenShift Container Platform. Pour plus d'informations, voir la section "VMware vSphere infrastructure requirements" dans la documentation d'OpenShift Container Platform. Vous pouvez également consulter la documentation de VMware sur les [versions matérielles des machines virtuelles](#).

3. Installez et configurez VMware Tools version 11.0.6 ou supérieure sur la VM Windows. Voir la [documentation de VMware Tools](#) pour plus d'informations.
4. Après avoir installé VMware Tools sur la VM Windows, vérifiez les points suivants :

- a. Le fichier **C:\ProgramData\VMware\VMware Tools\tools.conf** existe avec l'entrée suivante :

```
exclude-nics=
```

Si le fichier **tools.conf** n'existe pas, il est créé avec l'option **exclude-nics** décommentée et définie comme une valeur vide.

Cette entrée garantit que le vNIC cloné généré sur la VM Windows par la superposition hybride n'est pas ignoré.

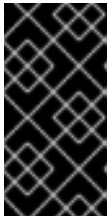
- b. La VM Windows a une adresse IP valide dans vCenter :

```
C:\N> ipconfig
```

- c. Le service Windows VMTools est en cours d'exécution :

```
PS C:\N> Get-Service -Name VMTools | Select Status, StartType
```

5. Installez et configurez le serveur OpenSSH sur la VM Windows. Voir la documentation de Microsoft sur l'[installation d'OpenSSH](#) pour plus de détails.
6. Configurez l'accès SSH pour un utilisateur administratif. Pour ce faire, consultez la documentation de Microsoft sur l'[utilisateur administratif](#).



IMPORTANT

La clé publique utilisée dans les instructions doit correspondre à la clé privée que vous créez plus tard dans l'espace de noms WMCO et qui contient votre secret. Voir la section "Configuration d'un secret pour l'opérateur de configuration Windows Machine" pour plus de détails.

7. Vous devez créer une nouvelle règle de pare-feu dans la VM Windows qui autorise les connexions entrantes pour les journaux de conteneurs. Exécutez la commande PowerShell suivante pour créer la règle de pare-feu sur le port TCP 10250 :

```
PS C:\N> New-NetFirewallRule -DisplayName \N "ContainerLogsPort" -LocalPort 10250 -Enabled True -Direction Inbound -Protocol TCP -Action Allow -EdgeTraversalPolicy Allow
```

8. Clonez la machine virtuelle Windows pour en faire une image réutilisable. Pour plus de détails, consultez la documentation VMware sur le [clonage d'une machine virtuelle existante](#).
9. Dans la VM Windows clonée, exécutez l'[outil Windows Sysprep](#):

```
C:\N> C:\NWindows\NSystem32\NSysprep\Nsysprep.exe /generalize /oobe /shutdown /unattend:<path_to_unattend.xml> 1
```

- 1** Indiquez le chemin d'accès à votre fichier **unattend.xml**.



NOTE

Le nombre de fois où vous pouvez exécuter la commande **sysprep** sur une image Windows est limité. Consultez la [documentation](#) de Microsoft pour plus d'informations.

Un exemple **unattend.xml** est fourni, qui contient toutes les modifications nécessaires pour l'OCMF. Vous devez modifier cet exemple ; il ne peut pas être utilisé directement.

Exemple 5.1. Exemple unattend.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="specialize">
    <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Microsoft-Windows-
International-Core" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS">
      <InputLocale>0409:00000409</InputLocale>
      <SystemLocale>en-US</SystemLocale>
      <UILanguage>en-US</UILanguage>
      <UILanguageFallback>en-US</UILanguageFallback>
      <UserLocale>en-US</UserLocale>
    </component>
    <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Microsoft-Windows-
Security-SPP-UX" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS">
      <SkipAutoActivation>>true</SkipAutoActivation>
    </component>
    <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Microsoft-Windows-
SQMApi" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS">
      <CEIPEnabled>0</CEIPEnabled>
    </component>
    <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Microsoft-Windows-
Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS">
      <ComputerName>winhost</ComputerName> ❶
    </component>
  </settings>
  <settings pass="oobeSystem">
    <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Microsoft-Windows-
Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS">
      <AutoLogon>
        <Enabled>>false</Enabled> ❷
      </AutoLogon>
      <OOBE>
        <HideEULAPage>>true</HideEULAPage>
        <HideLocalAccountScreen>>true</HideLocalAccountScreen>
        <HideOEMRegistrationScreen>>true</HideOEMRegistrationScreen>
      </OOBE>
    </component>
  </settings>
</unattend>
```

```

    <HideOnlineAccountScreens>true</HideOnlineAccountScreens>
    <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
    <NetworkLocation>Work</NetworkLocation>
    <ProtectYourPC>1</ProtectYourPC>
    <SkipMachineOOBE>true</SkipMachineOOBE>
    <SkipUserOOBE>true</SkipUserOOBE>
  </OOBE>
  <RegisteredOrganization>Organization</RegisteredOrganization>
  <RegisteredOwner>Owner</RegisteredOwner>
  <DisableAutoDaylightTimeSet>>false</DisableAutoDaylightTimeSet>
  <TimeZone>Eastern Standard Time</TimeZone>
  <UserAccounts>
    <AdministratorPassword>
      <Value>MyPassword</Value> 3
      <PlainText>>true</PlainText>
    </AdministratorPassword>
  </UserAccounts>
</component>
</settings>
</unattend>

```

- 1 Spécifiez le site **ComputerName**, qui doit respecter la [spécification des noms de Kubernetes](#). Ces spécifications s'appliquent également à la personnalisation du système d'exploitation invité effectuée sur le modèle résultant lors de la création de nouvelles VM.
- 2 Désactiver la connexion automatique pour éviter le problème de sécurité lié au fait de laisser un terminal ouvert avec des privilèges d'administrateur au démarrage. Il s'agit de la valeur par défaut, qui ne doit pas être modifiée.
- 3 Remplacez l'espace réservé **MyPassword** par le mot de passe du compte administrateur. Cela permet d'éviter que le compte Administrateur intégré ait un mot de passe vide par défaut. Suivez les [meilleures pratiques de Microsoft pour choisir un mot de passe](#).

Une fois l'outil Sysprep terminé, la VM Windows s'éteint. Vous ne devez plus utiliser ou mettre sous tension cette VM.

10. Convertir la VM Windows en [modèle dans vCenter](#).

5.3.2.1.1. Ressources supplémentaires

- [Configuration d'un secret pour l'opérateur de configuration de la machine Windows](#)
- [Exigences relatives à l'infrastructure VMware vSphere](#)

5.3.2.2. Activation de la communication avec le serveur API interne pour le WMCO sur vSphere

Le Windows Machine Config Operator (WMCO) télécharge les fichiers de configuration Ignition à partir du point de terminaison du serveur API interne. Vous devez activer la communication avec le serveur API interne pour que votre machine virtuelle Windows (VM) puisse télécharger les fichiers de configuration

Ignition et que le kubelet sur la VM configurée puisse uniquement communiquer avec le serveur API interne.

Conditions préalables

- Vous avez installé un cluster sur vSphere.

Procédure

- Ajoutez une nouvelle entrée DNS pour **api-int.<cluster_name>.<base_domain>** qui pointe vers l'URL du serveur API externe **api.<cluster_name>.<base_domain>**. Il peut s'agir d'un CNAME ou d'un enregistrement A supplémentaire.



NOTE

Le point de terminaison API externe a déjà été créé dans le cadre de l'installation initiale du cluster sur vSphere.

5.3.3. Exemple de YAML pour un objet Windows MachineSet sur vSphere

Cet exemple YAML définit un objet Windows **MachineSet** fonctionnant sur VMware vSphere sur lequel le Windows Machine Config Operator (WMCO) peut réagir.

```

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <windows_machine_set_name> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id> 3
      machine.openshift.io/cluster-api-machineset: <windows_machine_set_name> 4
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id> 5
        machine.openshift.io/cluster-api-machine-role: worker
        machine.openshift.io/cluster-api-machine-type: worker
        machine.openshift.io/cluster-api-machineset: <windows_machine_set_name> 6
        machine.openshift.io/os-id: Windows 7
    spec:
      metadata:
        labels:
          node-role.kubernetes.io/worker: "" 8
      providerSpec:
        value:
          apiVersion: vsphereprovider.openshift.io/v1beta1
          credentialsSecret:
            name: vsphere-cloud-credentials

```

```

diskGiB: 128 9
kind: VSphereMachineProviderSpec
memoryMiB: 16384
network:
  devices:
  - networkName: "<vm_network_name>" 10
numCPUs: 4
numCoresPerSocket: 1
snapshot: ""
template: <windows_vm_template_name> 11
userDataSecret:
  name: windows-user-data 12
workspace:
  datacenter: <vcenter_datacenter_name> 13
  datastore: <vcenter_datastore_name> 14
  folder: <vcenter_vm_folder_path> 15
  resourcePool: <vsphere_resource_pool> 16
  server: <vcenter_server_ip> 17

```

- 1 3 5** Spécifiez l'ID de l'infrastructure qui est basé sur l'ID du cluster que vous avez défini lorsque vous avez provisionné le cluster. Vous pouvez obtenir l'ID d'infrastructure en exécutant la commande suivante :

```
$ oc get -o jsonpath='{.status.infrastructureName}' infrastructure cluster
```

- 2 4 6** Spécifiez le nom de l'ensemble de machines de calcul Windows. Le nom de l'ensemble de machines de calcul ne peut pas comporter plus de 9 caractères, en raison de la manière dont les noms de machines sont générés dans vSphere.
- 7** Configurez l'ensemble de machines de calcul comme une machine Windows.
- 8** Configurez le nœud Windows en tant que machine de calcul.
- 9** Spécifiez la taille du disque de machine virtuelle vSphere (VMDK).



NOTE

Ce paramètre ne définit pas la taille de la partition Windows. Vous pouvez redimensionner la partition Windows en utilisant le fichier **unattend.xml** ou en créant l'image dorée de la machine virtuelle (VM) vSphere Windows avec la taille de disque requise.

- 10** Spécifiez le réseau VM vSphere sur lequel déployer l'ensemble de machines de calcul. Ce réseau VM doit se trouver là où d'autres machines de calcul Linux résident dans le cluster.
- 11** Indiquez le chemin complet du modèle de VM Windows vSphere à utiliser, par exemple **golden-images/windows-server-template**. Le nom doit être unique.



IMPORTANT

N'indiquez pas le modèle original de la VM. Le modèle de VM doit rester désactivé et doit être cloné pour les nouvelles machines Windows. Le démarrage du modèle de VM configure le modèle de VM en tant que VM sur la plateforme, ce qui l'empêche d'être utilisé comme modèle auquel les ensembles de machines de calcul peuvent appliquer des configurations.

- 12 Le site **windows-user-data** est créé par le WMCO lors de la configuration de la première machine Windows. Par la suite, le site **windows-user-data** est disponible pour tous les ensembles de machines de calcul suivants.
- 13 Spécifiez le centre de données vCenter sur lequel déployer l'ensemble de machines de calcul.
- 14 Spécifiez le Datastore vCenter sur lequel déployer l'ensemble de machines de calcul.
- 15 Indiquez le chemin d'accès au dossier vSphere VM dans vCenter, par exemple **/dc1/vm/user-inst-5ddjd**.
- 16 Facultatif : Spécifiez le pool de ressources vSphere pour vos machines virtuelles Windows.
- 17 Spécifiez l'IP du serveur vCenter ou le nom de domaine complet.

5.3.4. Création d'un ensemble de machines de calcul

En plus des ensembles de machines de calcul créés par le programme d'installation, vous pouvez créer vos propres ensembles pour gérer dynamiquement les ressources de calcul des machines pour les charges de travail spécifiques de votre choix.

Conditions préalables

- Déployer un cluster OpenShift Container Platform.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Créez un nouveau fichier YAML contenant l'échantillon de ressources personnalisées (CR) de l'ensemble de machines de calcul et nommé **<file_name>.yaml**.
Veillez à définir les valeurs des paramètres **<clusterID>** et **<role>**.
2. Facultatif : si vous n'êtes pas sûr de la valeur à définir pour un champ spécifique, vous pouvez vérifier un ensemble de machines de calcul existant dans votre cluster.
 - a. Pour répertorier les ensembles de machines de calcul de votre cluster, exécutez la commande suivante :

```
$ oc get machinesets -n openshift-machine-api
```

Exemple de sortie

```
NAME                                DESIRED  CURRENT  READY  AVAILABLE  AGE
```

```

agl030519-vplxk-worker-us-east-1a 1 1 1 1 55m
agl030519-vplxk-worker-us-east-1b 1 1 1 1 55m
agl030519-vplxk-worker-us-east-1c 1 1 1 1 55m
agl030519-vplxk-worker-us-east-1d 0 0 0 0 55m
agl030519-vplxk-worker-us-east-1e 0 0 0 0 55m
agl030519-vplxk-worker-us-east-1f 0 0 0 0 55m

```

- b. Pour afficher les valeurs d'une ressource personnalisée (CR) d'un ensemble de machines de calcul spécifique, exécutez la commande suivante :

```

$ oc get machineset <machineset_name> \
-n openshift-machine-api -o yaml

```

Exemple de sortie

```

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <infrastructure_id>-<role> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id>
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id>
        machine.openshift.io/cluster-api-machine-role: <role>
        machine.openshift.io/cluster-api-machine-type: <role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
    spec:
      providerSpec: 3
      ...

```

- 1** L'ID de l'infrastructure du cluster.
- 2** Une étiquette de nœud par défaut.



NOTE

Pour les clusters disposant d'une infrastructure fournie par l'utilisateur, un ensemble de machines de calcul ne peut créer que des machines de type **worker** et **infra**.

- 3** Les valeurs de la section **<providerSpec>** du CR de l'ensemble de machines de calcul sont spécifiques à la plate-forme. Pour plus d'informations sur les paramètres **<providerSpec>** dans le CR, consultez l'exemple de configuration du CR de l'ensemble de machines de calcul pour votre fournisseur.

3. Créez un CR **MachineSet** en exécutant la commande suivante :

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

Vérification

- Affichez la liste des ensembles de machines de calcul en exécutant la commande suivante :

```
$ oc get machineset -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-windows-worker-us-east-1a	1	1	1	1	11m
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0			55m
agl030519-vplxk-worker-us-east-1e	0	0			55m
agl030519-vplxk-worker-us-east-1f	0	0			55m

Lorsque le nouveau jeu de machines de calcul est disponible, les valeurs **DESIRED** et **CURRENT** correspondent. Si le jeu de machines de calcul n'est pas disponible, attendez quelques minutes et exécutez à nouveau la commande.

5.3.5. Ressources supplémentaires

- Pour plus d'informations sur la gestion des jeux de machines, voir la section *Machine management*.

5.4. CRÉATION D'UN OBJET WINDOWS MACHINESET SUR GCP

Vous pouvez créer un objet Windows **MachineSet** à des fins spécifiques dans votre cluster OpenShift Container Platform sur Google Cloud Platform (GCP). Par exemple, vous pouvez créer des ensembles de machines Windows d'infrastructure et des machines connexes afin de pouvoir déplacer des charges de travail Windows de support vers les nouvelles machines Windows.

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous utilisez un serveur Windows pris en charge comme image du système d'exploitation.

5.4.1. Vue d'ensemble de l'API Machine

L'API Machine est une combinaison de ressources primaires basées sur le projet Cluster API en amont et de ressources OpenShift Container Platform personnalisées.

Pour les clusters OpenShift Container Platform 4.12, l'API Machine effectue toutes les actions de gestion du provisionnement de l'hôte du nœud une fois l'installation du cluster terminée. Grâce à ce système, OpenShift Container Platform 4.12 offre une méthode de provisionnement élastique et dynamique au-dessus d'une infrastructure de cloud public ou privé.

Les deux principales ressources sont les suivantes

Machines

Unité fondamentale décrivant l'hôte d'un nœud. Une machine possède une spécification **providerSpec**, qui décrit les types de nœuds de calcul proposés par les différentes plateformes de cloud computing. Par exemple, un type de machine pour un nœud de travail sur Amazon Web Services (AWS) peut définir un type de machine spécifique et les métadonnées requises.

Jeux de machines

MachineSet les ressources sont des groupes de machines de calcul. Les ensembles de machines de calcul sont aux machines de calcul ce que les ensembles de répliques sont aux pods. Si vous avez besoin de plus de machines de calcul ou si vous devez les réduire, vous modifiez le champ **replicas** de la ressource **MachineSet** pour répondre à vos besoins en matière de calcul.



AVERTISSEMENT

Les machines du plan de contrôle ne peuvent pas être gérées par des ensembles de machines de calcul.

Les ensembles de machines du plan de contrôle fournissent des capacités de gestion pour les machines du plan de contrôle prises en charge qui sont similaires à celles que les ensembles de machines de calcul fournissent pour les machines de calcul.

Pour plus d'informations, voir "Gestion des machines du plan de contrôle".

Les ressources personnalisées suivantes ajoutent des capacités supplémentaires à votre cluster :

Machine autoscaler

La ressource **MachineAutoscaler** met automatiquement à l'échelle les machines de calcul dans un nuage. Vous pouvez définir les limites minimales et maximales de mise à l'échelle pour les nœuds d'un ensemble de machines de calcul spécifié, et l'autoscaler de machines maintient cette plage de nœuds.

L'objet **MachineAutoscaler** prend effet après l'existence d'un objet **ClusterAutoscaler**. Les ressources **ClusterAutoscaler** et **MachineAutoscaler** sont mises à disposition par l'objet **ClusterAutoscalerOperator**.

Cluster autoscaler

Cette ressource est basée sur le projet de cluster autoscaler en amont. Dans l'implémentation d'OpenShift Container Platform, elle est intégrée à l'API Machine en étendant l'API compute machine set. Vous pouvez utiliser le cluster autoscaler pour gérer votre cluster de la manière suivante :

- Définir des limites d'échelle à l'échelle du cluster pour les ressources telles que les cœurs, les nœuds, la mémoire et les GPU
- Définir la priorité afin que le cluster donne la priorité aux pods et que de nouveaux nœuds ne soient pas mis en ligne pour des pods moins importants

- Définir la politique de mise à l'échelle de manière à pouvoir augmenter les nœuds mais pas les diminuer

Bilan de santé de la machine

La ressource **MachineHealthCheck** détecte lorsqu'une machine n'est pas saine, la supprime et, sur les plates-formes prises en charge, crée une nouvelle machine.

Dans OpenShift Container Platform version 3.11, il n'était pas possible de déployer facilement une architecture multizone car le cluster ne gérait pas le provisionnement des machines. À partir de la version 4.1 d'OpenShift Container Platform, ce processus est plus facile. Chaque ensemble de machines de calcul est limité à une seule zone, de sorte que le programme d'installation envoie des ensembles de machines de calcul à travers les zones de disponibilité en votre nom. Ainsi, comme votre calcul est dynamique, et en cas de défaillance d'une zone, vous disposez toujours d'une zone pour rééquilibrer vos machines. Dans les régions Azure globales qui ne disposent pas de plusieurs zones de disponibilité, vous pouvez utiliser des ensembles de machines pour garantir une haute disponibilité. L'autoscaler assure l'équilibrage du meilleur effort pendant toute la durée de vie d'un cluster.

5.4.2. Exemple de YAML pour un objet Windows MachineSet sur GCP

Cet exemple de fichier YAML définit un objet Windows **MachineSet** fonctionnant sur Google Cloud Platform (GCP) que le Windows Machine Config Operator (WMCO) peut utiliser.

```

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <infrastructure_id>-windows-worker-<zone_suffix> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id> 3
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-windows-worker-<zone_suffix>
  4
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_id> 5
        machine.openshift.io/cluster-api-machine-role: worker
        machine.openshift.io/cluster-api-machine-type: worker
        machine.openshift.io/cluster-api-machineset: <infrastructure_id>-windows-worker-<zone_suffix>
  6
    machine.openshift.io/os-id: Windows 7
  spec:
    metadata:
      labels:
        node-role.kubernetes.io/worker: "" 8
    providerSpec:
      value:
        apiVersion: machine.openshift.io/v1beta1
        canIPForward: false
        credentialsSecret:

```

```

name: gcp-cloud-credentials
deletionProtection: false
disks:
- autoDelete: true
  boot: true
  image: <windows_server_image> 9
  sizeGb: 128
  type: pd-ssd
kind: GCPMachineProviderSpec
machineType: n1-standard-4
networkInterfaces:
- network: <infrastructure_id>-network 10
  subnetwork: <infrastructure_id>-worker-subnet
projectID: <project_id> 11
region: <region> 12
serviceAccounts:
- email: <infrastructure_id>-w@<project_id>.iam.gserviceaccount.com
  scopes:
  - https://www.googleapis.com/auth/cloud-platform
tags:
- <infrastructure_id>-worker
userDataSecret:
  name: windows-user-data 13
zone: <zone> 14

```

1 3 5 10 Spécifiez l'ID de l'infrastructure qui est basé sur l'ID du cluster que vous avez défini lorsque vous avez approvisionné le cluster. Vous pouvez obtenir l'ID d'infrastructure en exécutant la commande suivante :

```
$ oc get -o jsonpath='{.status.infrastructureName}' infrastructure cluster
```

2 4 6 Spécifiez l'ID de l'infrastructure, l'étiquette du travailleur et le suffixe de la zone (tel que **a**).

7 Configurez le jeu de machines comme une machine Windows.

8 Configurez le nœud Windows en tant que machine de calcul.

9 Indiquez le chemin d'accès complet à une image d'une version prise en charge de Windows Server.

11 Spécifiez le projet GCP dans lequel ce cluster a été créé.

12 Spécifiez la région du GCP, par exemple **us-central1**.

13 Créé par le WMCO lorsqu'il configure la première machine Windows. Ensuite, le site **windows-user-data** est disponible pour tous les jeux de machines suivants.

14 Indiquez la zone de la région choisie, par exemple **us-central1-a**.

5.4.3. Création d'un ensemble de machines de calcul

En plus des ensembles de machines de calcul créés par le programme d'installation, vous pouvez créer vos propres ensembles pour gérer dynamiquement les ressources de calcul des machines pour les charges de travail spécifiques de votre choix.

Conditions préalables

- Déployer un cluster OpenShift Container Platform.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Créez un nouveau fichier YAML contenant l'échantillon de ressources personnalisées (CR) de l'ensemble de machines de calcul et nommé **<file_name>.yaml**.
Veillez à définir les valeurs des paramètres **<clusterID>** et **<role>**.
2. Facultatif : si vous n'êtes pas sûr de la valeur à définir pour un champ spécifique, vous pouvez vérifier un ensemble de machines de calcul existant dans votre cluster.
 - a. Pour répertorier les ensembles de machines de calcul de votre cluster, exécutez la commande suivante :

```
$ oc get machinesets -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0			55m
agl030519-vplxk-worker-us-east-1e	0	0			55m
agl030519-vplxk-worker-us-east-1f	0	0			55m

- b. Pour afficher les valeurs d'une ressource personnalisée (CR) d'un ensemble de machines de calcul spécifique, exécutez la commande suivante :

```
$ oc get machineset <machineset_name> \
-n openshift-machine-api -o yaml
```

Exemple de sortie

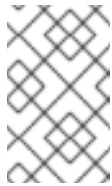
```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id> 1
  name: <infrastructure_id>-<role> 2
  namespace: openshift-machine-api
spec:
  replicas: 1
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id>
      machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
  template:
```

```

metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id>
    machine.openshift.io/cluster-api-machine-role: <role>
    machine.openshift.io/cluster-api-machine-type: <role>
    machine.openshift.io/cluster-api-machineset: <infrastructure_id>-<role>
spec:
  providerSpec: 3
  ...

```

- 1 L'ID de l'infrastructure du cluster.
- 2 Une étiquette de nœud par défaut.



NOTE

Pour les clusters disposant d'une infrastructure fournie par l'utilisateur, un ensemble de machines de calcul ne peut créer que des machines de type **worker** et **infra**.

- 3 Les valeurs de la section **<providerSpec>** du CR de l'ensemble de machines de calcul sont spécifiques à la plate-forme. Pour plus d'informations sur les paramètres **<providerSpec>** dans le CR, consultez l'exemple de configuration du CR de l'ensemble de machines de calcul pour votre fournisseur.

3. Créez un CR **MachineSet** en exécutant la commande suivante :

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

Vérification

- Affichez la liste des ensembles de machines de calcul en exécutant la commande suivante :

```
$ oc get machineset -n openshift-machine-api
```

Exemple de sortie

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
agl030519-vplxk-infra-us-east-1a	1	1	1	1	11m
agl030519-vplxk-worker-us-east-1a	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1b	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1c	1	1	1	1	55m
agl030519-vplxk-worker-us-east-1d	0	0			55m
agl030519-vplxk-worker-us-east-1e	0	0			55m
agl030519-vplxk-worker-us-east-1f	0	0			55m

Lorsque le nouveau jeu de machines de calcul est disponible, les valeurs **DESIRED** et **CURRENT** correspondent. Si le jeu de machines de calcul n'est pas disponible, attendez quelques minutes et exécutez à nouveau la commande.

5.4.4. Ressources supplémentaires

- Pour plus d'informations sur la gestion des jeux de machines, voir [Vue d'ensemble de la gestion des machines](#).

CHAPITRE 6. PLANIFICATION DES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS

Vous pouvez planifier des charges de travail Windows sur des nœuds de calcul Windows.



NOTE

Le WMCO n'est pas pris en charge dans les clusters qui utilisent un [proxy à l'échelle du cluster](#) car le WMCO n'est pas en mesure d'acheminer le trafic via la connexion proxy pour les charges de travail.

Conditions préalables

- Vous avez installé Windows Machine Config Operator (WMCO) à l'aide d'Operator Lifecycle Manager (OLM).
- Vous utilisez un conteneur Windows comme image du système d'exploitation.
- Vous avez créé un ensemble de machines de calcul Windows.

6.1. PLACEMENT DES NACELLES DE FENÊTRES

Avant de déployer vos charges de travail Windows dans le cluster, vous devez configurer la planification de votre nœud Windows afin que les pods soient affectés correctement. Étant donné qu'une machine héberge votre nœud Windows, celui-ci est géré de la même manière qu'un nœud basé sur Linux. De même, la planification d'un pod Windows sur le nœud Windows approprié s'effectue de la même manière, à l'aide de mécanismes tels que les taints, les tolérances et les sélecteurs de nœuds.

Avec plusieurs systèmes d'exploitation et la possibilité d'exécuter plusieurs variantes de Windows OS dans le même cluster, vous devez mapper vos pods Windows à une variante de base de Windows OS à l'aide d'un objet **RuntimeClass**. Par exemple, si vous avez plusieurs nœuds Windows fonctionnant sur différentes versions de conteneurs Windows Server, le cluster pourrait planifier vos pods Windows sur une variante incompatible du système d'exploitation Windows. Les objets **RuntimeClass** doivent être configurés pour chaque variante de système d'exploitation Windows sur votre cluster. L'utilisation d'un objet **RuntimeClass** est également recommandée si vous ne disposez que d'une seule variante du système d'exploitation Windows dans votre cluster.

Pour plus d'informations, voir la documentation de Microsoft sur la [compatibilité des versions de l'hôte et du conteneur](#).



IMPORTANT

L'image de base du conteneur doit correspondre à la version du système d'exploitation Windows et au numéro de version en cours d'exécution sur le nœud où le conteneur doit être planifié.

De même, si vous mettez à niveau les nœuds Windows d'une version à une autre, par exemple en passant de 20H2 à 2022, vous devez mettre à niveau l'image de base de votre conteneur pour qu'elle corresponde à la nouvelle version. Pour plus d'informations, voir [Compatibilité des versions des conteneurs Windows](#).

Ressources supplémentaires

- [Contrôler le placement des pods à l'aide de l'ordonnanceur](#)

- Contrôle du placement de pods à l'aide de taches de nœuds
- Placer des pods sur des nœuds spécifiques en utilisant des sélecteurs de nœuds

6.2. CRÉATION D'UN OBJET `RuntimeClass` POUR ENCAPSULER LES MÉCANISMES DE PLANIFICATION

L'utilisation d'un objet **RuntimeClass** simplifie l'utilisation de mécanismes de planification tels que les taints et les tolérances ; vous déployez une classe d'exécution qui encapsule vos taints et vos tolérances, puis vous l'appliquez à vos pods pour les planifier sur le nœud approprié. La création d'une classe d'exécution est également nécessaire dans les clusters qui prennent en charge plusieurs variantes de systèmes d'exploitation.

Procédure

1. Créer un fichier YAML de l'objet **RuntimeClass**. Par exemple, `runtime-class.yaml`:

```
apiVersion: node.k8s.io/v1beta1
kind: RuntimeClass
metadata:
  name: <runtime_class_name> 1
handler: 'runhcs-wcow-process'
scheduling:
  nodeSelector: 2
    kubernetes.io/os: 'windows'
    kubernetes.io/arch: 'amd64'
    node.kubernetes.io/windows-build: '10.0.17763'
  tolerations: 3
    - effect: NoSchedule
      key: os
      operator: Equal
      value: "Windows"
```

- 1 Spécifiez le nom de l'objet **RuntimeClass**, qui est défini dans les pods que vous souhaitez voir gérés par cette classe d'exécution.
- 2 Spécifiez les étiquettes qui doivent être présentes sur les nœuds qui prennent en charge cette classe d'exécution. Les modules utilisant cette classe d'exécution ne peuvent être planifiés que sur un nœud correspondant à ce sélecteur. Le sélecteur de nœud de la classe d'exécution est fusionné avec le sélecteur de nœud existant du module. Tout conflit empêche la programmation du module sur le nœud.
- 3 Spécifiez les tolérances à ajouter aux modules, à l'exclusion des doublons, fonctionnant avec cette classe d'exécution lors de l'admission. Cela combine l'ensemble des nœuds tolérés par le module et la classe d'exécution.

2. Créer l'objet **RuntimeClass**:

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f runtime-class.yaml
```

3. Appliquez l'objet **RuntimeClass** à votre pod pour vous assurer qu'il est planifié sur la variante appropriée du système d'exploitation :

```

apiVersion: v1
kind: Pod
metadata:
  name: my-windows-pod
spec:
  runtimeClassName: <runtime_class_name> 1
  ...

```

- 1 Spécifiez la classe d'exécution pour gérer l'ordonnancement de votre module.

6.3. EXEMPLE DE DÉPLOIEMENT D'UNE CHARGE DE TRAVAIL DE CONTENEUR WINDOWS

Vous pouvez déployer des charges de travail de conteneurs Windows dans votre cluster dès que vous disposez d'un nœud de calcul Windows.



NOTE

Cet exemple de déploiement est fourni à titre de référence uniquement.

Exemple d'objet Service

```

apiVersion: v1
kind: Service
metadata:
  name: win-webserver
labels:
  app: win-webserver
spec:
  ports:
    # the port that this service should serve on
    - port: 80
      targetPort: 80
  selector:
    app: win-webserver
  type: LoadBalancer

```

Exemple d'objet Deployment

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: win-webserver
  name: win-webserver
spec:
  selector:
    matchLabels:
      app: win-webserver

```

```

replicas: 1
template:
  metadata:
    labels:
      app: win-webserver
      name: win-webserver
  spec:
    tolerations:
      - key: "os"
        value: "Windows"
        Effect: "NoSchedule"
    containers:
      - name: windowswebserver
        image: mcr.microsoft.com/windows/servercore:ltsc2019
        imagePullPolicy: IfNotPresent
        command:
          - powershell.exe
          - -command
            - $listener = New-Object System.Net.HttpListener; $listener.Prefixes.Add('http://*:80/');
              $listener.Start(); Write-Host('Listening at http://*:80/'); while ($listener.IsListening) { $context =
                $listener.GetContext(); $response = $context.Response; $content='<html><body><H1>Red Hat
                OpenShift + Windows Container Workloads</H1></body></html>'; $buffer =
                [System.Text.Encoding]::UTF8.GetBytes($content); $response.ContentLength64 = $buffer.Length;
                $response.OutputStream.Write($buffer, 0, $buffer.Length); $response.Close(); };
        securityContext:
          runAsNonRoot: false
        windowsOptions:
          runAsUserName: "ContainerAdministrator"
    nodeSelector:
      kubernetes.io/os: windows

```



NOTE

Si vous utilisez l'image de conteneur **mcr.microsoft.com/powershell:<tag>**, vous devez définir la commande comme **pwsh.exe**. Si vous utilisez l'image de conteneur **mcr.microsoft.com/windows/servercore:<tag>**, vous devez définir la commande comme **powershell.exe**. Pour plus d'informations, consultez la documentation de Microsoft.

6.4. MISE À L'ÉCHELLE MANUELLE D'UN ENSEMBLE DE MACHINES DE CALCUL

Pour ajouter ou supprimer une instance d'une machine dans un ensemble de machines de calcul, vous pouvez mettre à l'échelle manuellement l'ensemble de machines de calcul.

Ce guide s'applique aux installations d'infrastructure entièrement automatisées et fournies par l'installateur. Les installations d'infrastructure personnalisées et fournies par l'utilisateur n'ont pas d'ensembles de machines de calcul.

Conditions préalables

- Installer un cluster OpenShift Container Platform et la ligne de commande **oc**.
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Affichez les ensembles de machines de calcul qui se trouvent dans le cluster en exécutant la commande suivante :

```
$ oc get machinesets -n openshift-machine-api
```

Les ensembles de machines de calcul sont répertoriés sous la forme de **<clusterid>-worker-
<aws-region-az>**.

2. Affichez les machines de calcul qui se trouvent dans le cluster en exécutant la commande suivante :

```
$ oc get machine -n openshift-machine-api
```

3. Définissez l'annotation sur la machine de calcul que vous souhaitez supprimer en exécutant la commande suivante :

```
$ oc annotate machine/<machine_name> -n openshift-machine-api  
machine.openshift.io/delete-machine="true"
```

4. Mettez à l'échelle l'ensemble de machines de calcul en exécutant l'une des commandes suivantes :

```
$ oc scale --replicas=2 machineset <machineset> -n openshift-machine-api
```

Ou bien :

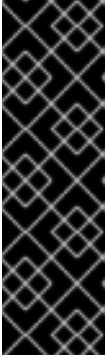
```
$ oc edit machineset <machineset> -n openshift-machine-api
```

ASTUCE

Vous pouvez également appliquer le YAML suivant pour mettre à l'échelle l'ensemble des machines de calcul :

```
apiVersion: machine.openshift.io/v1beta1  
kind: MachineSet  
metadata:  
  name: <machineset>  
  namespace: openshift-machine-api  
spec:  
  replicas: 2
```

Vous pouvez augmenter ou diminuer le nombre de machines de calcul. Il faut quelques minutes pour que les nouvelles machines soient disponibles.



IMPORTANT

Par défaut, le contrôleur de machine tente de drainer le nœud soutenu par la machine jusqu'à ce qu'il y parvienne. Dans certaines situations, comme dans le cas d'un budget de perturbation de pods mal configuré, l'opération de vidange peut ne pas aboutir. Si l'opération de vidange échoue, le contrôleur de machine ne peut pas procéder au retrait de la machine.

Vous pouvez éviter de vider le nœud en annotant **`machine.openshift.io/exclude-node-draining`** dans une machine spécifique.

Vérification

- Vérifiez la suppression de la machine prévue en exécutant la commande suivante :

```
$ oc get machines
```

CHAPITRE 7. MISES À NIVEAU DES NŒUDS WINDOWS

Vous pouvez vous assurer que vos nœuds Windows disposent des dernières mises à jour en mettant à jour le Windows Machine Config Operator (WMCO).

7.1. MISE À NIVEAU DE L'OPÉRATEUR DE CONFIGURATION DE LA MACHINE WINDOWS

Lorsqu'une nouvelle version de l'opérateur de configuration de la machine Windows (WMCO) compatible avec la version actuelle du cluster est publiée, l'opérateur est mis à niveau en fonction du canal de mise à niveau et de la stratégie d'approbation de l'abonnement avec lesquels il a été installé lors de l'utilisation du gestionnaire du cycle de vie de l'opérateur (OLM). La mise à niveau de WMCO entraîne la mise à niveau des composants Kubernetes dans la machine Windows.



NOTE

Si vous passez à une nouvelle version de l'OCMW et que vous voulez utiliser la surveillance de grappes, vous devez avoir le label **openshift.io/cluster-monitoring=true** dans l'espace de noms de l'OCMW. Si vous ajoutez le label à un espace de noms WMCO préexistant et que des nœuds Windows sont déjà configurés, redémarrez le pod WMCO pour permettre l'affichage des graphiques de surveillance.

Dans le cas d'une mise à niveau non perturbatrice, l'OCMW met fin aux machines Windows configurées par la version précédente de l'OCMW et les recrée à l'aide de la version actuelle. Pour ce faire, il supprime l'objet **Machine**, ce qui entraîne la vidange et la suppression du nœud Windows. Pour faciliter une mise à niveau, l'OCMW ajoute une annotation de version à tous les nœuds configurés. Au cours d'une mise à niveau, une erreur dans l'annotation de la version entraîne la suppression et la recréation d'une machine Windows. Pour minimiser les interruptions de service lors d'une mise à niveau, l'OCMF ne met à jour qu'une seule machine Windows à la fois.



IMPORTANT

Le WMCO est uniquement responsable de la mise à jour des composants Kubernetes, et non des mises à jour du système d'exploitation Windows. Vous fournissez l'image Windows lors de la création des VMs ; vous êtes donc responsable de la fourniture d'une image mise à jour. Vous pouvez fournir une image Windows mise à jour en modifiant la configuration de l'image dans la spécification **MachineSet**.

Pour plus d'informations sur les mises à niveau d'opérateurs à l'aide du gestionnaire du cycle de vie des opérateurs (OLM), voir [Mise à jour des opérateurs installés](#).

CHAPITRE 8. UTILISATION D'INSTANCES WINDOWS BYOH (BRING-YOUR-OWN-HOST) EN TANT QUE NŒUDS

Bring-Your-Own-Host (BYOH) permet aux utilisateurs de réutiliser les VMs Windows Server et de les apporter à OpenShift Container Platform. Les instances Windows BYOH sont utiles aux utilisateurs qui cherchent à atténuer les perturbations majeures en cas de panne d'un serveur Windows.

8.1. CONFIGURATION D'UNE INSTANCE WINDOWS BYOH

La création d'une instance Windows BYOH nécessite la création d'une carte de configuration dans l'espace de noms Windows Machine Config Operator (WMCO).

Conditions préalables

Toutes les instances Windows qui doivent être attachées au cluster en tant que nœud doivent remplir les conditions suivantes :

- L'instance doit se trouver sur le même réseau que les nœuds de travail Linux dans le cluster.
- Le port 22 doit être ouvert et un serveur SSH doit fonctionner.
- L'interpréteur de [commandes](#) par défaut du serveur SSH doit être l' [interpréteur de commandes Windows](#) ou `cmd.exe`.
- Le port 10250 doit être ouvert pour la collecte des journaux.
- Un utilisateur administrateur est présent avec la clé privée utilisée dans l'ensemble de secrets comme clé SSH autorisée.
- Si vous créez une instance Windows BYOH pour un cluster AWS d'infrastructure fournie par l'installateur (IPI), vous devez ajouter une balise à l'instance AWS qui correspond à la valeur **spec.template.spec.value.tag** dans l'ensemble de machines de calcul pour vos nœuds de travail. Par exemple, **kubernetes.io/cluster/<cluster_id>: owned** ou **kubernetes.io/cluster/<cluster_id>: shared**.
- Si vous créez une instance Windows BYOH sur vSphere, la communication avec le serveur API interne doit être activée.
- Le nom d'hôte de l'instance doit respecter les exigences de l'étiquette DNS [RFC 1123](#), qui comprend les normes suivantes :
 - Ne contient que des caractères alphanumériques minuscules ou "-".
 - Commence par un caractère alphanumérique.
 - Se termine par un caractère alphanumérique.



NOTE

Les instances Windows déployées par le WMCO sont configurées avec le moteur d'exécution de conteneur containerd. Étant donné que le WMCO installe et gère le runtime, il est recommandé de ne pas installer manuellement containerd sur les nœuds.

Procédure

1. Créez un ConfigMap nommé **windows-instances** dans l'espace de noms WMCO qui décrit les instances Windows à ajouter.



NOTE

Formatez chaque entrée dans la section de données de la carte de configuration en utilisant l'adresse comme clé et en formatant la valeur comme **username=<username>**.

Exemple de carte de configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: windows-instances
  namespace: openshift-windows-machine-config-operator
data:
  10.1.42.1: |- 1
    username=Administrator 2
  instance.example.com: |-
    username=core
```

- 1 L'adresse que le WMCO utilise pour atteindre l'instance via SSH, soit un nom DNS, soit une adresse IPv4. Un enregistrement DNS PTR doit exister pour cette adresse. Il est recommandé d'utiliser un nom DNS avec votre instance BYOH si votre organisation utilise DHCP pour attribuer les adresses IP. Si ce n'est pas le cas, vous devez mettre à jour le ConfigMap de **windows-instances** chaque fois qu'une nouvelle adresse IP est attribuée à l'instance.
- 2 Le nom de l'utilisateur administrateur créé dans les conditions préalables.

8.2. SUPPRESSION DES INSTANCES WINDOWS BYOH

Vous pouvez supprimer les instances BYOH attachées au cluster en supprimant l'entrée de l'instance dans la carte de configuration. La suppression d'une instance rétablit l'état dans lequel elle se trouvait avant d'être ajoutée au cluster. Les journaux et les artefacts d'exécution des conteneurs ne sont pas ajoutés à ces instances.

Pour qu'une instance soit proprement supprimée, elle doit être accessible avec la clé privée actuelle fournie à WMCO. Par exemple, pour supprimer l'instance **10.1.42.1** de l'exemple précédent, la carte de configuration serait modifiée comme suit :

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: windows-instances
  namespace: openshift-windows-machine-config-operator
data:
  instance.example.com: |-
    username=core
```

La suppression de **windows-instances** est considérée comme une demande de déconstruction de toutes les instances Windows ajoutées en tant que nœuds.

CHAPITRE 9. SUPPRESSION DES NŒUDS WINDOWS

Vous pouvez supprimer un nœud Windows en supprimant sa machine Windows hôte.

9.1. SUPPRESSION D'UNE MACHINE SPÉCIFIQUE

Vous pouvez supprimer une machine spécifique.



IMPORTANT

Ne supprimez pas une machine du plan de contrôle à moins que votre cluster n'utilise un jeu de machines du plan de contrôle.

Conditions préalables

- Installer un cluster OpenShift Container Platform.
- Installez le CLI OpenShift (**oc**).
- Connectez-vous à **oc** en tant qu'utilisateur disposant de l'autorisation **cluster-admin**.

Procédure

1. Affichez les machines qui se trouvent dans le cluster en exécutant la commande suivante :

```
$ oc get machine -n openshift-machine-api
```

La sortie de la commande contient une liste de machines au format **<clusterid>-<role>-<cloud_region>**.

2. Identifiez la machine que vous souhaitez supprimer.
3. Supprimez la machine en exécutant la commande suivante :

```
$ oc delete machine <machine> -n openshift-machine-api
```



IMPORTANT

Par défaut, le contrôleur de machine tente de drainer le nœud soutenu par la machine jusqu'à ce qu'il y parvienne. Dans certaines situations, comme dans le cas d'un budget de perturbation de pods mal configuré, l'opération de vidange peut ne pas aboutir. Si l'opération de vidange échoue, le contrôleur de machine ne peut pas procéder au retrait de la machine.

Vous pouvez éviter de vider le nœud en annotant **machine.openshift.io/exclude-node-draining** dans une machine spécifique.

Si la machine que vous supprimez appartient à un ensemble de machines, une nouvelle machine est immédiatement créée pour satisfaire le nombre de répliques spécifié.

CHAPITRE 10. DÉSACTIVER LES CHARGES DE TRAVAIL DES CONTENEURS WINDOWS

Vous pouvez désactiver la capacité d'exécuter des charges de travail de conteneur Windows en désinstallant le Windows Machine Config Operator (WMCO) et en supprimant l'espace de noms qui a été ajouté par défaut lors de l'installation du WMCO.

10.1. DÉINSTALLATION DE L'OPÉRATEUR WINDOWS MACHINE CONFIG

Vous pouvez désinstaller le Windows Machine Config Operator (WMCO) de votre cluster.

Conditions préalables

- Supprimez les objets Windows **Machine** qui hébergent vos charges de travail Windows.

Procédure

1. À partir de la page **Operators → OperatorHub**, utilisez la boîte **Filter by keyword** pour rechercher **Red Hat Windows Machine Config Operator**.
2. Cliquez sur la tuile **Red Hat Windows Machine Config Operator**. La tuile Opérateur indique qu'elle est installée.
3. Dans la page du descripteur **Windows Machine Config Operator**, cliquez sur **Uninstall**.

10.2. SUPPRESSION DE L'ESPACE DE NOMS WINDOWS MACHINE CONFIG OPERATOR

Vous pouvez supprimer l'espace de noms généré par défaut pour le Windows Machine Config Operator (WMCO).

Conditions préalables

- Le WMCO est retiré de votre cluster.

Procédure

1. Supprimez toutes les charges de travail Windows créées dans l'espace de noms **openshift-windows-machine-config-operator**:

```
$ oc delete --all pods --namespace=openshift-windows-machine-config-operator
```

2. Vérifiez que tous les pods de l'espace de noms **openshift-windows-machine-config-operator** sont supprimés ou signalent un état d'arrêt :

```
$ oc get pods --namespace openshift-windows-machine-config-operator
```

3. Supprimer l'espace de noms **openshift-windows-machine-config-operator**:

```
$ oc delete namespace openshift-windows-machine-config-operator
```

Ressources supplémentaires

- [Suppression d'opérateurs d'une grappe](#)
- [Suppression des nœuds Windows](#)