Plate-forme de conteneurs OpenShift 4.12

Sauvegarde et restauration

Sauvegarder et restaurer votre cluster OpenShift Container Platform

Last Updated: 2023-04-26

Plate-forme de conteneurs OpenShift 4.12 Sauvegarde et restauration

Sauvegarder et restaurer votre cluster OpenShift Container Platform

Notice légale

Résumé

Ce document fournit des instructions pour la sauvegarde des données de votre cluster et pour la récupération à partir de différents scénarios de catastrophe.

Table des matières

CHAPITRE 1. SAUVEGARDE ET RESTAURATION 3
1.1. OPÉRATIONS DE SAUVEGARDE ET DE RESTAURATION DU PLAN DE CONTRÔLE 3
1.2. OPÉRATIONS DE SAUVEGARDE ET DE RESTAURATION DES APPLICATIONS 3

CHAPITRE 2. ARRÊTER LE CLUSTER DE MANIÈRE ÉLÉGANTE 6
2.1. CONDITIONS PRÉALABLES 6
2.2. ARRÊT DU CLUSTER 6
2.3. RESSOURCES COMPLÉMENTAIRES 8

CHAPITRE 3. REDÉMARRER LE CLUSTER AVEC ÉLÉGANCE 9
3.1. CONDITIONS PRÉALABLES 9
3.2. REDÉMARRAGE DU CLUSTER 9

CHAPITRE 4. SAUVEGARDE ET RESTAURATION DES APPLICATIONS 12
4.1. NOTES DE MISE À JOUR DE L'OADP 12
4.2. FONCTIONNALITÉS ET PLUGINS DE L'OADP 13
4.3. INSTALLATION ET CONFIGURATION DE L'OADP 17
4.4. SAUVEGARDE ET RESTAURATION 63
4.5. DÉPANNAGE 79
4.6. API UTILISÉES AVEC L'OADP 88
4.7. CARACTÉRISTIQUES ET FONCTIONNALITÉS AVANCÉES DE L'OADP 94

CHAPITRE 5. SAUVEGARDE ET RESTAURATION DU PLAN DE CONTRÔLE 97
5.1. SAUVEGARDE DE ETCD 97
5.2. REMPLACEMENT D'UN MEMBRE ETCD MALSAIN 99
5.3. SAUVEGARDE ET RESTAURATION D'ETCD SUR UN CLUSTER HÉBERGÉ 125
5.4. REPRISE APRÈS SINISTRE 128

CHAPITRE 1. SAUVEGARDE ET RESTAURATION

1.1. OPÉRATIONS DE SAUVEGARDE ET DE RESTAURATION DU PLAN DE CONTRÔLE

En tant qu'administrateur de cluster, vous pouvez avoir besoin d'arrêter un cluster OpenShift Container Platform pendant une période et de le redémarrer plus tard. Le redémarrage d'un cluster peut être motivé par la nécessité d'effectuer des opérations de maintenance sur un cluster ou par la volonté de réduire les coûts de ressources. Dans OpenShift Container Platform, vous pouvez effectuer un arrêt gracieux d'un cluster afin de pouvoir le redémarrer facilement plus tard.

Vous devez sauvegarder les données etcd avant d'arrêter un cluster ; etcd est le magasin clé-valeur d'OpenShift Container Platform, qui conserve l'état de tous les objets de ressources. Une sauvegarde etcd joue un rôle crucial dans la reprise après sinistre. Dans OpenShift Container Platform, vous pouvez également remplacer un membre etcd malsain.

Lorsque vous souhaitez remettre votre cluster en marche, redémarrez-le avec élégance.

NOTE

Les certificats d'une grappe expirent un an après la date d'installation. Vous pouvez arrêter une grappe et vous attendre à ce qu'elle redémarre sans problème tant que les certificats sont encore valides. Bien que la grappe récupère automatiquement les certificats de plan de contrôle expirés, vous devez toujours approuver les demandes de signature de certificat (CSR).

Vous pouvez rencontrer plusieurs situations dans lesquelles OpenShift Container Platform ne fonctionne pas comme prévu, par exemple :

- Vous avez un cluster qui n'est pas fonctionnel après le redémarrage en raison de conditions inattendues, telles que la défaillance d'un nœud ou des problèmes de connectivité réseau.
- Vous avez supprimé par erreur un élément essentiel du cluster.
- Vous avez perdu la majorité des hôtes du plan de contrôle, ce qui entraîne la perte du quorum etcd.

Vous pouvez toujours vous remettre d'une situation de désastre en restaurant votre cluster à son état précédent en utilisant les snapshots etcd sauvegardés.

Ressources complémentaires

- Protection du quorum à l'aide de crochets de cycle de vie de la machine

1.2. OPÉRATIONS DE SAUVEGARDE ET DE RESTAURATION DES APPLICATIONS

En tant qu'administrateur de cluster, vous pouvez sauvegarder et restaurer des applications fonctionnant sur OpenShift Container Platform en utilisant l'API OpenShift pour la protection des données (OADP).

OADP sauvegarde et restaure les ressources Kubernetes et les images internes, à la granularité d'un espace de noms, en utilisant la version de Velero appropriée à la version d'OADP que vous installez,

conformément au tableau de la section Téléchargement de l'outil Velero CLI. OADP sauvegarde et restaure les volumes persistants (PV) à l'aide d'instantanés ou de Restic. Pour plus d'informations, voir Fonctionnalités de l'OADP.

1.2.1. Exigences de l'OADP

L'OADP a les exigences suivantes :

- Vous devez être connecté en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous devez disposer d'un stockage d'objets pour stocker les sauvegardes, tel que l'un des types de stockage suivants :
  - OpenShift Data Foundation
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - Stockage d'objets compatible S3

NOTE

Si vous souhaitez utiliser la sauvegarde CSI sur l'OCP 4.11 et les versions ultérieures, installez l'OADP 1.1.*x*.

OADP 1.0.*x* ne prend pas en charge la sauvegarde CSI sur OCP 4.11 et les versions ultérieures. OADP 1.0.*x* inclut Velero 1.7.*x* et attend le groupe API **snapshot.storage.k8s.io/v1beta1**, qui n'est pas présent sur l'OCP 4.11 et les versions ultérieures.

IMPORTANT

L'API **CloudStorage** pour le stockage S3 est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

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

- Pour sauvegarder des PV à l'aide d'instantanés, vous devez disposer d'un stockage en nuage doté d'une API d'instantanés native ou prenant en charge les instantanés de l'interface de stockage de conteneurs (CSI), tels que les fournisseurs suivants :
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform

- Stockage en nuage CSI compatible avec les instantanés, tel que Ceph RBD ou Ceph FS

NOTE

Si vous ne souhaitez pas sauvegarder les PV à l'aide d'instantanés, vous pouvez utiliser Restic, qui est installé par défaut par l'opérateur OADP.

1.2.2. Sauvegarde et restauration des applications

Vous sauvegardez les applications en créant une **Backup** ressource personnalisée (CR). Vous pouvez configurer les options de sauvegarde suivantes :

- Crochets de sauvegarde pour exécuter des commandes avant ou après l'opération de sauvegarde
- Sauvegardes programmées
- Sauvegardes Restic

Vous restaurez les applications en créant un **Restore** CR. Vous pouvez configurer les crochets de restauration pour qu'ils exécutent des commandes dans les conteneurs init ou dans le conteneur d'application pendant l'opération de restauration.

CHAPITRE 2. ARRÊTER LE CLUSTER DE MANIÈRE ÉLÉGANTE

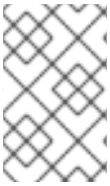
Ce document décrit le processus d'arrêt gracieux de votre cluster. Vous pouvez avoir besoin d'arrêter temporairement votre cluster pour des raisons de maintenance ou pour économiser des ressources.

2.1. CONDITIONS PRÉALABLES

- Effectuer une sauvegarde d'etcd avant d'arrêter le cluster.

2.2. ARRÊT DU CLUSTER

Vous pouvez arrêter votre cluster de manière gracieuse afin qu'il puisse être redémarré ultérieurement.

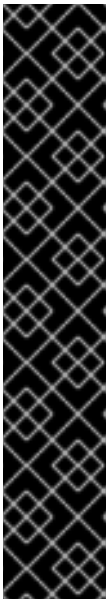


NOTE

Vous pouvez arrêter une grappe jusqu'à un an après la date d'installation et vous attendre à ce qu'elle redémarre sans problème. Après un an à compter de la date d'installation, les certificats de la grappe expirent.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez effectué une sauvegarde etcd.



IMPORTANT

Il est important de faire une sauvegarde d'etcd avant d'effectuer cette procédure afin que votre cluster puisse être restauré si vous rencontrez des problèmes lors du redémarrage du cluster.

Par exemple, les conditions suivantes peuvent entraîner un dysfonctionnement du cluster redémarré :

- corruption des données etcd lors de l'arrêt
- Défaillance d'un nœud due au matériel
- Problèmes de connectivité du réseau

Si votre cluster ne se rétablit pas, suivez les étapes pour rétablir l'état précédent du cluster.

Procédure

1. Si vous arrêtez le cluster pour une période prolongée, déterminez la date d'expiration des certificats.

```
$ oc -n openshift-kube-apiserver-operator get secret kube-apiserver-to-kubelet-signer -o
jsonpath='{.metadata.annotations.auth\.openshift\.io/certificate-not-after}'
```

Exemple de sortie

```
2022-08-05T14:37:50Zuser@user:~ $ 1
```

1 Pour garantir un redémarrage en douceur du cluster, prévoyez de le redémarrer au plus tard à la date spécifiée. Lors du redémarrage du cluster, il se peut que vous deviez approuver manuellement les demandes de signature de certificat (CSR) en attente pour récupérer les certificats des kubelets.

2. Arrêtez tous les nœuds du cluster. Vous pouvez le faire à partir de la console web de votre fournisseur de cloud ou exécuter la boucle suivante :

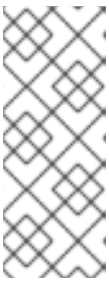
```
for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}') ; do oc debug
node/${node} -- chroot /host shutdown -h 1 ; done 1
```

1 **-h 1** indique la durée, en minutes, de ce processus avant l'arrêt des nœuds du plan de contrôle. Pour les clusters à grande échelle de 10 nœuds ou plus, définissez une durée de 10 minutes ou plus afin de vous assurer que tous les nœuds de calcul ont le temps de s'arrêter en premier.

Exemple de sortie

```
Starting pod/ip-10-0-130-169us-east-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Shutdown scheduled for Mon 2021-09-13 09:36:17 UTC, use 'shutdown -c' to cancel.

Removing debug pod ...
Starting pod/ip-10-0-150-116us-east-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Shutdown scheduled for Mon 2021-09-13 09:36:29 UTC, use 'shutdown -c' to cancel.
```

L'arrêt des nœuds à l'aide de l'une de ces méthodes permet aux pods de se terminer de manière élégante, ce qui réduit le risque de corruption des données.



NOTE

Ajustez le temps d'arrêt pour qu'il soit plus long pour les grappes à grande échelle :

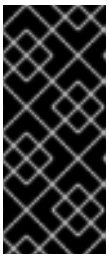
```
$ for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}'); do oc
debug node/${node} -- chroot /host shutdown -h 10; done
```

NOTE

Il n'est pas nécessaire de vider les nœuds du plan de contrôle des pods standard livrés avec OpenShift Container Platform avant l'arrêt.

Les administrateurs de clusters sont chargés de garantir un redémarrage propre de leurs propres charges de travail après le redémarrage du cluster. Si vous avez vidangé des nœuds du plan de contrôle avant l'arrêt en raison de charges de travail personnalisées, vous devez marquer les nœuds du plan de contrôle comme planifiables pour que le cluster soit à nouveau fonctionnel après le redémarrage.

3. Désactivez toutes les dépendances du cluster qui ne sont plus nécessaires, telles que le stockage externe ou le serveur LDAP. Veillez à consulter la documentation de votre fournisseur avant de procéder à cette opération.



IMPORTANT

Si vous avez déployé votre cluster sur une plateforme de fournisseur de cloud, n'arrêtez pas, ne suspendez pas et ne supprimez pas les ressources cloud associées. Si vous supprimez les ressources cloud d'une machine virtuelle suspendue, OpenShift Container Platform risque de ne pas se restaurer correctement.

2.3. RESSOURCES COMPLÉMENTAIRES

- Redémarrer le cluster avec élégance
- Rétablissement d'un état antérieur de la grappe

CHAPITRE 3. REDÉMARRER LE CLUSTER AVEC ÉLÉGANCE

Ce document décrit le processus de redémarrage de votre cluster après un arrêt gracieux.

Même si le cluster est censé être fonctionnel après le redémarrage, il se peut qu'il ne se rétablisse pas en raison de conditions inattendues, par exemple :

- corruption des données etcd lors de l'arrêt
- Défaillance d'un nœud due au matériel
- Problèmes de connectivité du réseau

Si votre cluster ne se rétablit pas, suivez les étapes pour rétablir l'état précédent du cluster.

3.1. CONDITIONS PRÉALABLES

- Vous avez arrêté votre cluster de manière élégante .

3.2. REDÉMARRAGE DU CLUSTER

Vous pouvez redémarrer votre cluster après qu'il ait été arrêté de manière gracieuse.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Cette procédure suppose que vous avez arrêté le cluster de manière gracieuse.

Procédure

1. Activez toutes les dépendances du cluster, telles que le stockage externe ou un serveur LDAP.
2. Démarrer toutes les machines du cluster.
   Utilisez la méthode appropriée à votre environnement cloud pour démarrer les machines, par exemple à partir de la console web de votre fournisseur de cloud.

   Attendez environ 10 minutes avant de continuer à vérifier l'état des nœuds du plan de contrôle.
3. Vérifier que tous les nœuds du plan de contrôle sont prêts.

   ```
   $ oc get nodes -l node-role.kubernetes.io/master
   ```

   Les nœuds du plan de contrôle sont prêts si l'état est **Ready**, comme le montre la sortie suivante :

   ```
   NAME                          STATUS   ROLES    AGE   VERSION
   ip-10-0-168-251.ec2.internal   Ready    master   75m   v1.25.0
   ip-10-0-170-223.ec2.internal   Ready    master   75m   v1.25.0
   ip-10-0-211-16.ec2.internal    Ready    master   75m   v1.25.0
   ```
4. Si les nœuds du plan de contrôle sont prêts pour *not*, vérifiez s'il y a des demandes de signature de certificat (CSR) en attente qui doivent être approuvées.

a. Obtenir la liste des CSR actuels :

```
$ oc get csr
```

b. Examinez les détails d'un CSR pour vérifier qu'il est valide :

```
oc describe csr <csr_name> 1
```

1 **<csr_name>** est le nom d'un CSR figurant dans la liste des CSR actuels.

c. Approuver chaque RSE valide :

```
$ oc adm certificate approve <csr_name>
```

5. Une fois que les nœuds du plan de contrôle sont prêts, vérifiez que tous les nœuds de travail sont prêts.

```
$ oc get nodes -l node-role.kubernetes.io/worker
```

Les nœuds de travail sont prêts si le statut est **Ready**, comme le montre la sortie suivante :

```
NAME                         STATUS   ROLES    AGE   VERSION
ip-10-0-179-95.ec2.internal   Ready    worker   64m   v1.25.0
ip-10-0-182-134.ec2.internal  Ready    worker   64m   v1.25.0
ip-10-0-250-100.ec2.internal  Ready    worker   64m   v1.25.0
```

6. Si les nœuds de travail sont prêts pour *not*, vérifiez s'il y a des demandes de signature de certificat (CSR) en attente qui doivent être approuvées.

a. Obtenir la liste des CSR actuels :

```
$ oc get csr
```

b. Examinez les détails d'un CSR pour vérifier qu'il est valide :

```
oc describe csr <csr_name> 1
```

1 **<csr_name>** est le nom d'un CSR figurant dans la liste des CSR actuels.

c. Approuver chaque RSE valide :

```
$ oc adm certificate approve <csr_name>
```

7. Vérifiez que le cluster a démarré correctement.

a. Vérifiez qu'il n'y a pas d'opérateurs de cluster dégradés.

```
$ oc get clusteroperators
```

Vérifiez qu'il n'y a pas d'opérateurs de cluster dont la condition **DEGRADED** est définie sur **True**.

```
NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.12.0    True        False         False      59m
cloud-credential                           4.12.0    True        False         False      85m
cluster-autoscaler                         4.12.0    True        False         False      73m
config-operator                            4.12.0    True        False         False      73m
console                                    4.12.0    True        False         False      62m
csi-snapshot-controller                    4.12.0    True        False         False      66m
dns                                        4.12.0    True        False         False      76m
etcd                                       4.12.0    True        False         False      76m
...
```

b. Vérifiez que tous les nœuds sont dans l'état **Ready**:

```
$ oc get nodes
```

Vérifiez que l'état de tous les nœuds est **Ready**.

```
NAME                           STATUS   ROLES    AGE   VERSION
ip-10-0-168-251.ec2.internal   Ready    master   82m   v1.25.0
ip-10-0-170-223.ec2.internal   Ready    master   82m   v1.25.0
ip-10-0-179-95.ec2.internal    Ready    worker   70m   v1.25.0
ip-10-0-182-134.ec2.internal   Ready    worker   70m   v1.25.0
ip-10-0-211-16.ec2.internal    Ready    master   82m   v1.25.0
ip-10-0-250-100.ec2.internal   Ready    worker   69m   v1.25.0
```

Si le cluster n'a pas démarré correctement, il se peut que vous deviez le restaurer à l'aide d'une sauvegarde etcd.

Ressources complémentaires

- Voir Restauration d'un état antérieur du cluster pour savoir comment utiliser une sauvegarde etcd pour restaurer si votre cluster n'a pas réussi à se rétablir après un redémarrage.

CHAPITRE 4. SAUVEGARDE ET RESTAURATION DES APPLICATIONS

4.1. NOTES DE MISE À JOUR DE L'OADP

Les notes de version d'OpenShift API for Data Protection (OADP) décrivent les nouvelles fonctionnalités et améliorations, les fonctionnalités obsolètes, les recommandations de produit, les problèmes connus et les problèmes résolus.

4.1.1. Notes de publication de l'OADP 1.1.2

Les notes de mise à jour de l'OADP 1.1.2 comprennent des recommandations sur le produit, une liste des bogues corrigés et des descriptions des problèmes connus.

4.1.1.1. Recommandations de produits

VolSync

Pour préparer la mise à jour de VolSync 0.5.1 vers la dernière version disponible sur le canal VolSync **stable**, vous devez ajouter cette annotation dans l'espace de noms **openshift-adp** en exécutant la commande suivante :

```
$ oc annotate --overwrite namespace/openshift-adp volsync.backube/privileged-movers='true'
```

Velero

Dans cette version, Velero est passé de la version 1.9.2 à la version 1.9.5.

Restic

Dans cette version, Restic est passé de la version 0.13.1 à la version 0.14.0.

4.1.1.2. Bugs corrigés

Les bogues suivants ont été corrigés dans cette version :

- OADP-1150
- OADP-290
- OADP-1056

4.1.1.3. Problèmes connus

Cette version comporte les problèmes connus suivants :

- Actuellement, l'OADP ne prend pas en charge la sauvegarde et la restauration des volumes AWS EFS à l'aide de restic dans Velero (**OADP-778**).
- Les sauvegardes CSI peuvent échouer en raison d'une limitation Ceph de **VolumeSnapshotContent** snapshots par PVC.
  Vous pouvez créer plusieurs instantanés de la même revendication de volume persistant (PVC), mais vous ne pouvez pas planifier la création périodique d'instantanés :

- Pour CephFS, vous pouvez créer jusqu'à 100 instantanés par PVC. (OADP-804)
- Pour RADOS Block Device (RBD), vous pouvez créer jusqu'à 512 instantanés pour chaque PVC. (OADP-975)

Pour plus d'informations, voir Instantanés de volume .

4.1.2. Notes de mise à jour de l'OADP 1.1.1

Les notes de mise à jour de l'OADP 1.1.1 comprennent des recommandations sur le produit et des descriptions des problèmes connus.

4.1.2.1. Recommandations de produits

Avant d'installer OADP 1.1.1, il est recommandé d'installer VolSync 0.5.1 ou de le mettre à jour.

4.1.2.2. Problèmes connus

Cette version comporte les problèmes connus suivants :

- Actuellement, l'OADP ne prend pas en charge la sauvegarde et la restauration des volumes AWS EFS à l'aide de restic dans Velero (OADP-778).
- Les sauvegardes CSI peuvent échouer en raison d'une limitation Ceph de **VolumeSnapshotContent** snapshots par PVC.
  Vous pouvez créer plusieurs instantanés de la même revendication de volume persistant (PVC), mais vous ne pouvez pas planifier la création périodique d'instantanés :
  - Pour CephFS, vous pouvez créer jusqu'à 100 instantanés par PVC.
  - Pour RADOS Block Device (RBD), vous pouvez créer jusqu'à 512 snapshots pour chaque PVC. (OADP-804) et (OADP-975)
    Pour plus d'informations, voir Instantanés de volume .

4.2. FONCTIONNALITÉS ET PLUGINS DE L'OADP

Les fonctionnalités d'OpenShift API for Data Protection (OADP) offrent des options de sauvegarde et de restauration des applications.

Les plugins par défaut permettent à Velero de s'intégrer à certains fournisseurs de cloud et de sauvegarder et restaurer les ressources d'OpenShift Container Platform.

4.2.1. Caractéristiques de l'OADP

OpenShift API for Data Protection (OADP) prend en charge les fonctionnalités suivantes :

Sauvegarde

Vous pouvez sauvegarder toutes les ressources de votre cluster ou filtrer les ressources par type, espace de noms ou étiquette.
L'OADP sauvegarde les objets Kubernetes et les images internes en les enregistrant en tant que fichier d'archive sur le stockage d'objets. L'OADP sauvegarde les volumes persistants (PV) en créant des instantanés à l'aide de l'API native d'instantané du nuage ou de l'interface de stockage de conteneurs (CSI). Pour les fournisseurs de cloud qui ne prennent pas en charge les instantanés, l'OADP sauvegarde les ressources et les données des PV avec Restic.

Restaurer

Vous pouvez restaurer des ressources et des PV à partir d'une sauvegarde. Vous pouvez restaurer tous les objets d'une sauvegarde ou filtrer les objets restaurés par espace de noms, PV ou étiquette.

Calendrier

Vous pouvez planifier des sauvegardes à des intervalles déterminés.

Crochets

Vous pouvez utiliser des hooks pour exécuter des commandes dans un conteneur sur un pod, par exemple, **fsfreeze** pour geler un système de fichiers. Vous pouvez configurer un hook pour qu'il s'exécute avant ou après une sauvegarde ou une restauration. Les hooks de restauration peuvent être exécutés dans un conteneur init ou dans le conteneur d'application.

4.2.2. Plugins OADP

L'API OpenShift pour la protection des données (OADP) fournit des plugins Velero par défaut qui sont intégrés avec les fournisseurs de stockage pour prendre en charge les opérations de sauvegarde et d'instantané. Vous pouvez créer des plugins personnalisés basés sur les plugins Velero.

OADP fournit également des plugins pour les sauvegardes de ressources OpenShift Container Platform, les sauvegardes de ressources OpenShift Virtualization et les instantanés de l'interface de stockage de conteneurs (CSI).

Tableau 4.1. Plugins OADP

Plugin OADP	Fonction	Lieu de stockage
aws	Sauvegarde et restauration des objets Kubernetes.	AWS S3
	Sauvegarde et restauration de volumes à l'aide d'instantanés.	AWS EBS
azure	Sauvegarde et restauration des objets Kubernetes.	Stockage Blob de Microsoft Azure
	Sauvegarde et restauration de volumes à l'aide d'instantanés.	Disques gérés Microsoft Azure
gcp	Sauvegarde et restauration des objets Kubernetes.	Stockage dans le nuage de Google
	Sauvegarde et restauration de volumes à l'aide d'instantanés.	Disques Google Compute Engine
openshift	Sauvegarde et restaure les ressources de OpenShift Container Platform. [1]	Magasin d'objets
kubevirt	Sauvegarde et restauration des ressources de virtualisation OpenShift. [2]	Magasin d'objets

Plugin OADP	Fonction	Lieu de stockage
csi	Sauvegarde et restauration de volumes avec des instantanés CSI. [3]	Stockage en nuage prenant en charge les instantanés CSI

1. Obligatoire.
2. Les disques des machines virtuelles sont sauvegardés à l'aide d'instantanés CSI ou Restic.
3. Le plugin **csi** utilise l'API Velero CSI beta snapshot.

4.2.3. A propos des plugins OADP Velero

Vous pouvez configurer deux types de plugins lorsque vous installez Velero :

- Plugins de fournisseurs de cloud par défaut
- Plugins personnalisés

Les deux types de plugins sont facultatifs, mais la plupart des utilisateurs configurent au moins un plugin de fournisseur de cloud.

4.2.3.1. Plugins par défaut du fournisseur de cloud Velero

Vous pouvez installer l'un des plugins Velero suivants lorsque vous configurez le fichier **oadp_v1alpha1_dpa.yaml** pendant le déploiement :

- **aws** (Amazon Web Services)
- **gcp** (Google Cloud Platform)
- **azure** (Microsoft Azure)
- **openshift** (plugin OpenShift Velero)
- **csi** (Interface de stockage de conteneurs)
- **kubevirt** (KubeVirt)

Vous spécifiez les plugins par défaut souhaités dans le fichier **oadp_v1alpha1_dpa.yaml** lors du déploiement.

Exemple de fichier

Le fichier **.yaml** suivant installe les plugins **openshift**, **aws**, **azure** et **gcp**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    velero:
```

```
    defaultPlugins:
    - openshift
    - aws
    - azure
    - gcp
```

4.2.3.2. Plugins Velero personnalisés

Vous pouvez installer un plugin Velero personnalisé en spécifiant les plugins **image** et **name** lorsque vous configurez le fichier **oadp_v1alpha1_dpa.yaml** pendant le déploiement.

Vous spécifiez les plugins personnalisés souhaités dans le fichier **oadp_v1alpha1_dpa.yaml** lors du déploiement.

Exemple de fichier

Le fichier **.yaml** suivant installe les plugins par défaut **openshift**, **azure**, et **gcp** ainsi qu'un plugin personnalisé portant le nom **custom-plugin-example** et l'image **quay.io/example-repo/custom-velero-plugin**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: dpa-sample
spec:
 configuration:
   velero:
     defaultPlugins:
     - openshift
     - azure
     - gcp
     customPlugins:
     - name: custom-plugin-example
       image: quay.io/example-repo/custom-velero-plugin
```

4.2.4. Prise en charge de l'OADP pour les systèmes IBM Power et IBM zSystems

OpenShift API for Data Protection (OADP) est neutre en termes de plateforme. Les informations qui suivent concernent uniquement IBM Power et IBM zSystems.

OADP 1.1.0 a été testé avec succès contre OpenShift Container Platform 4.11 pour les systèmes IBM Power et IBM zSystems. Les sections suivantes donnent des informations sur les tests et le support pour OADP 1.1.0 en termes d'emplacements de sauvegarde pour ces systèmes.

4.2.4.1. Prise en charge de l'OADP pour les sites de sauvegarde cibles utilisant IBM Power

IBM Power fonctionnant avec OpenShift Container Platform 4.11 et 4.12, et OpenShift API for Data Protection (OADP) 1.1.2 a été testé avec succès contre une cible d'emplacement de sauvegarde AWS S3. Bien que le test n'ait impliqué qu'une cible AWS S3, Red Hat prend en charge l'exécution d'IBM Power avec OpenShift Container Platform 4.11 et 4.12, et OADP 1.1.2 contre toutes les cibles d'emplacement de sauvegarde S3 non-AWS également.

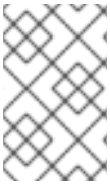
4.2.4.2. Tests et assistance OADP pour les sites de sauvegarde cibles utilisant des systèmes IBM z

IBM zSystems fonctionnant avec OpenShift Container Platform 4.11 et 4.12, et OpenShift API for Data Protection (OADP) 1.1.2 a été testé avec succès contre une cible d'emplacement de sauvegarde AWS S3. Bien que le test n'ait impliqué qu'une cible AWS S3, Red Hat prend en charge l'exécution d'IBM zSystems avec OpenShift Container Platform 4.11 et 4.12, et OADP 1.1.2 contre toutes les cibles d'emplacement de sauvegarde S3 non-AWS également.

4.3. INSTALLATION ET CONFIGURATION DE L'OADP

4.3.1. A propos de l'installation de l'OADP

En tant qu'administrateur de cluster, vous installez l'API OpenShift pour la protection des données (OADP) en installant l'opérateur OADP. L'opérateur OADP installe Velero 1.9.



NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Pour sauvegarder les ressources Kubernetes et les images internes, vous devez disposer d'un stockage d'objets comme emplacement de sauvegarde, tel que l'un des types de stockage suivants :

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Passerelle d'objets multicloud
- Stockage d'objets compatible S3, tel que Noobaa ou Minio

IMPORTANT

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

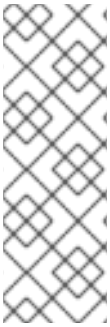
Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

Vous pouvez sauvegarder des volumes persistants (PV) à l'aide d'instantanés ou de Restic.

Pour sauvegarder des PV à l'aide d'instantanés, vous devez disposer d'un fournisseur de cloud qui prend en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI), comme l'un des fournisseurs de cloud suivants :

- Amazon Web Services

- Microsoft Azure
- Google Cloud Platform
- Fournisseur de services en nuage CSI compatible avec les instantanés, tel qu'OpenShift Data Foundation



NOTE

Si vous souhaitez utiliser la sauvegarde CSI sur l'OCP 4.11 et les versions ultérieures, installez l'OADP 1.1.*x*.

OADP 1.0.*x* ne prend pas en charge la sauvegarde CSI sur OCP 4.11 et les versions ultérieures. OADP 1.0.*x* inclut Velero 1.7.*x* et attend le groupe API **snapshot.storage.k8s.io/v1beta1**, qui n'est pas présent sur l'OCP 4.11 et les versions ultérieures.

Si votre fournisseur de cloud computing ne prend pas en charge les instantanés ou si votre stockage est de type NFS, vous pouvez sauvegarder des applications avec des sauvegardes Restic sur un stockage d'objets.

Vous créez un site **Secret** par défaut, puis vous installez l'application de protection des données.

4.3.1.1. Configurer NooBaa pour la reprise après sinistre sur OpenShift Data Foundation

Si vous utilisez le stockage en cluster pour votre bucket NooBaa **backupStorageLocation** sur OpenShift Data Foundation, configurez NooBaa en tant que magasin d'objets externe.

AVERTISSEMENT

Si NooBaa n'est pas configuré comme un magasin d'objets externe, les sauvegardes risquent de ne pas être disponibles.

Procédure

- Configurez NooBaa en tant que magasin d'objets externe comme décrit dans Ajouter des ressources de stockage pour l'hybride ou le Multicloud.

Ressources complémentaires

- Aperçu des emplacements de sauvegarde et de snapshot dans la documentation Velero

4.3.1.2. À propos des canaux de mise à jour de l'OADP

Lorsque vous installez un opérateur OADP, vous choisissez un canal *update channel*. Ce canal détermine les mises à jour de l'opérateur OADP et de Velero que vous recevez. Vous pouvez changer de canal à tout moment.

Il existe trois canaux de mise à jour :

- Le canal **stable** contient les dernières mises à jour mineures (y-stream updates) et les correctifs (z-stream updates) de OADP ClusterServiceVersion `. Au fur et à mesure de la publication de chaque nouvelle version, le site **ClusterServiceVersion** de l'opérateur OADP sera complété par le dernier correctif mineur disponible.

- Le canal **stable-1.0** contient **oadp.v1.0.*z*** la version la plus récente de l'OADP 1.0 **ClusterServiceVersion**.

- Le canal **stable-1.1** contient **oadp.v1.1.*z*** la version la plus récente de l'OADP 1.1 **ClusterServiceVersion**.

Which update channel is right for you?

- Choisissez le canal de mise à jour **stable** pour installer la dernière version stable de l'OADP et recevoir les mises à jour mineures et les correctifs. Si vous choisissez ce canal, vous recevrez toutes les mises à jour y-stream et toutes les mises à jour z-stream de la version *x.y.z*.

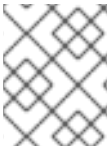
- Choisissez le canal de mise à jour **stable-1.*y*** update channel pour installer OADP 1. *y* et continuer à recevoir les correctifs. Si vous choisissez ce canal, vous recevrez tous les correctifs z-stream pour la version 1.*y.z*.

When must you switch update channels?

- Si vous avez installé OADP 1.*y* et que vous souhaitez recevoir des correctifs uniquement pour ce flux, vous devez passer du canal de mise à jour **stable** au canal de mise à jour **stable-1.*y*** canal de mise à jour. Vous recevrez alors tous les correctifs du flux z pour la version 1.*y.z*.

- Si vous avez installé OADP 1.0, que vous souhaitez passer à OADP 1.1 et recevoir des correctifs uniquement pour OADP 1.1, vous devez passer du canal de mise à jour **stable-1.0** au canal de mise à jour **stable-1.1**. Vous recevrez alors tous les correctifs z-stream pour la version 1.1. *z*.

- Si vous avez installé OADP 1.*y*, avec *y* supérieur à 0, et que vous souhaitez passer à OADP 1.0, vous devez *uninstall* votre opérateur OADP et le réinstaller en utilisant le canal de mise à jour **stable-1.0**. Vous recevrez alors tous les correctifs z-stream pour la version 1.0. *z*.



NOTE

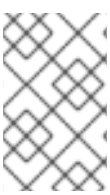
Vous ne pouvez pas passer de l'OADP 1.*y* à l'OADP 1.0 en changeant de canal de mise à jour. Vous devez désinstaller l'opérateur et le réinstaller.

Ressources complémentaires

- Version du service de cluster

4.3.2. Installation et configuration d'OpenShift API for Data Protection avec Amazon Web Services

Vous installez OpenShift API for Data Protection (OADP) avec Amazon Web Services (AWS) en installant l'opérateur OADP. L'opérateur installe Velero 1.9.



NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Vous configurez AWS pour Velero, créez une adresse **Secret** par défaut, puis installez l'application de protection des données.

Pour installer l'OADP Operator dans un environnement réseau restreint, vous devez d'abord désactiver les sources OperatorHub par défaut et mettre en miroir le catalogue Operator. Voir Utilisation d'Operator Lifecycle Manager sur des réseaux restreints pour plus de détails.

4.3.2.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec des privilèges **cluster-admin**.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.
3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

4.3.2.2. Configuration d'Amazon Web Services

Vous configurez Amazon Web Services (AWS) pour OpenShift API for Data Protection (OADP).

Conditions préalables

- Le logiciel AWS CLI doit être installé.

Procédure

1. Définir la variable **BUCKET**:

```
$ BUCKET=<your_bucket>
```

2. Définir la variable **REGION**:

```
$ REGION=<your_region>
```

3. Créez un seau AWS S3 :

```
$ aws s3api create-bucket \
    --bucket $BUCKET \
    --region $REGION \
    --create-bucket-configuration LocationConstraint=$REGION 1
```

1 **us-east-1** ne prend pas en charge **LocationConstraint**. Si votre région est **us-east-1**, omettez **--create-bucket-configuration LocationConstraint=$REGION**.

4. Créer un utilisateur IAM :

```
aws iam create-user --user-name velero 1
```

1 Si vous souhaitez utiliser Velero pour sauvegarder plusieurs clusters avec plusieurs buckets S3, créez un nom d'utilisateur unique pour chaque cluster.

5. Créer un fichier **velero-policy.json**:

```
$ cat > velero-policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:CreateTags",
                "ec2:CreateVolume",
                "ec2:CreateSnapshot",
                "ec2:DeleteSnapshot"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "arn:aws:s3:::${BUCKET}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::${BUCKET}"
            ]
        }
```

```
    ]
}
EOF
```

6. Attachez les politiques pour donner à l'utilisateur **velero** les permissions minimales nécessaires :

```
$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero \
  --policy-document file://velero-policy.json
```

7. Créer une clé d'accès pour l'utilisateur **velero**:

```
$ aws iam create-access-key --user-name velero
```

Exemple de sortie

```
{
  "AccessKey": {
        "UserName": "velero",
        "Status": "Active",
        "CreateDate": "2017-07-31T22:24:41.576Z",
        "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>,
        "AccessKeyId": <AWS_ACCESS_KEY_ID>
  }
}
```

8. Créer un fichier **credentials-velero**:

```
$ cat << EOF > ./credentials-velero
[default]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
EOF
```

Vous utilisez le fichier **credentials-velero** pour créer un objet **Secret** pour AWS avant d'installer l'application de protection des données.

4.3.2.3. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.

IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

4.3.2.3.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

Le nom par défaut du site **Secret** est **cloud-credentials**.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

4.3.2.3.2. Création de profils pour différentes informations d'identification

Si vos emplacements de sauvegarde et d'instantané utilisent des identifiants différents, vous créez des profils distincts dans le fichier **credentials-velero**.

Ensuite, vous créez un objet **Secret** et spécifiez les profils dans la ressource personnalisée (CR) **DataProtectionApplication**.

Procédure

1. Créez un fichier **credentials-velero** avec des profils distincts pour les emplacements de sauvegarde et d'instantané, comme dans l'exemple suivant :

```
[backupStorage]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>

[volumeSnapshot]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
```

2. Créer un objet **Secret** avec le fichier **credentials-velero**:

```
oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero (1)
```

3. Ajoutez les profils au CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
        config:
```

```
        region: us-east-1
        profile: "backupStorage"
      credential:
        key: cloud
        name: cloud-credentials
  snapshotLocations:
    - name: default
      velero:
        provider: aws
        config:
          region: us-west-2
          profile: "volumeSnapshot"
```

4.3.2.4. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

4.3.2.4.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 1 Spécifier le sélecteur de nœud à fournir à Velero podSpec

4.3.2.4.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

1. Indiquez la chaîne du certificat de l'autorité de certification codée en base64.
2. La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

4.3.2.5. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.
- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).

- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials**.
- Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials**, qui contient des profils distincts pour les informations d'identification de l'emplacement de sauvegarde et de l'emplacement d'instantané.

NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - openshift (1)
        - aws
    restic:
      enable: true (2)
      podConfig:
        nodeSelector: <node selector> (3)
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name> (4)
          prefix: <prefix> (5)
        config:
          region: <region>
          profile: "default"
        credential:
          key: cloud
          name: cloud-credentials (6)
  snapshotLocations: (7)
    - name: default
```

```
    velero:
      provider: aws
      config:
        region: <region> 8
        profile: "default"
```

1. Le plugin **openshift** est obligatoire.
2. Définissez **false** si vous souhaitez désactiver l'installation de Restic. Restic déploie un ensemble de démons, ce qui signifie que chaque nœud de travailleur a des pods **Restic** en cours d'exécution. Vous configurez Restic pour les sauvegardes en ajoutant **spec.defaultVolumesToRestic: true** au CR **Backup**.
3. Spécifie le sélecteur de nœud à fournir à Restic podSpec.
4. Spécifiez un bac comme emplacement de stockage des sauvegardes. Si le bac n'est pas un bac dédié aux sauvegardes Velero, vous devez spécifier un préfixe.
5. Spécifiez un préfixe pour les sauvegardes Velero, par exemple, **velero**, si le seau est utilisé à des fins multiples.
6. Indiquez le nom de l'objet **Secret** que vous avez créé. Si vous ne spécifiez pas cette valeur, le nom par défaut, **cloud-credentials**, est utilisé. Si vous spécifiez un nom personnalisé, celui-ci est utilisé pour l'emplacement de sauvegarde.
7. Il n'est pas nécessaire de spécifier un emplacement d'instantané si vous utilisez des instantanés CSI ou Restic pour sauvegarder des PV.
8. L'emplacement de l'instantané doit être situé dans la même région que les PV.

4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                                 READY   UP-TO-DATE   AVAILABLE   AGE
```

```
deployment.apps/oadp-operator-controller-manager   1/1   1     1     2m9s
deployment.apps/velero                             1/1   1     1     96s

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

4.3.2.5.1. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) Ajouter le plugin par défaut **csi**.

4.3.3. Installation et configuration de l'API OpenShift pour la protection des données avec Microsoft Azure

Vous installez OpenShift API for Data Protection (OADP) avec Microsoft Azure en installant l'opérateur OADP. L'opérateur installe Velero 1.9.

NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Vous configurez Azure pour Velero, créez un site par défaut **Secret**, puis installez l'application de protection des données.

Pour installer l'OADP Operator dans un environnement réseau restreint, vous devez d'abord désactiver les sources OperatorHub par défaut et mettre en miroir le catalogue Operator. Voir Utilisation d'Operator Lifecycle Manager sur des réseaux restreints pour plus de détails.

4.3.3.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec des privilèges **cluster-admin**.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.
3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

4.3.3.2. Configuration de Microsoft Azure

Vous configurez un Microsoft Azure pour l'OpenShift API for Data Protection (OADP).

Conditions préalables

- Le logiciel Azure CLI doit être installé.

Procédure

1. Connectez-vous à Azure :

```
$ az login
```

2. Définir la variable **AZURE_RESOURCE_GROUP**:

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

3. Créez un groupe de ressources Azure :

```
$ az group create -n $AZURE_RESOURCE_GROUP --location CentralUS 1
```

1 Précisez votre lieu de résidence.

4. Définir la variable **AZURE_STORAGE_ACCOUNT_ID**:

```
$ AZURE_STORAGE_ACCOUNT_ID="velero$(uuidgen | cut -d '-' -f5 | tr '[A-Z]' '[a-z]')"
```

5. Créez un compte de stockage Azure :

```
$ az storage account create \
    --name $AZURE_STORAGE_ACCOUNT_ID \
```

```
  --resource-group $AZURE_RESOURCE_GROUP \
  --sku Standard_GRS \
  --encryption-services blob \
  --https-only true \
  --kind BlobStorage \
  --access-tier Hot
```

6. Définir la variable **BLOB_CONTAINER**:

```
$ BLOB_CONTAINER=velero
```

7. Créez un conteneur de stockage Azure Blob :

```
$ az storage container create \
  -n $BLOB_CONTAINER \
  --public-access off \
  --account-name $AZURE_STORAGE_ACCOUNT_ID
```

8. Obtenir la clé d'accès au compte de stockage :

```
$ AZURE_STORAGE_ACCOUNT_ACCESS_KEY=`az storage account keys list \
  --account-name $AZURE_STORAGE_ACCOUNT_ID \
  --query "[?keyName == 'key1'].value" -o tsv`
```

9. Créez un rôle personnalisé doté des autorisations minimales requises :

```
AZURE_ROLE=Velero
az role definition create --role-definition '{
   "Name": "'$AZURE_ROLE'",
   "Description": "Velero related permissions to perform backups, restores and deletions",
   "Actions": [
       "Microsoft.Compute/disks/read",
       "Microsoft.Compute/disks/write",
       "Microsoft.Compute/disks/endGetAccess/action",
       "Microsoft.Compute/disks/beginGetAccess/action",
       "Microsoft.Compute/snapshots/read",
       "Microsoft.Compute/snapshots/write",
       "Microsoft.Compute/snapshots/delete",
       "Microsoft.Storage/storageAccounts/listkeys/action",
       "Microsoft.Storage/storageAccounts/regeneratekey/action"
   ],
   "AssignableScopes": ["/subscriptions/'$AZURE_SUBSCRIPTION_ID'"]
   }'
```

10. Créer un fichier **credentials-velero**:

```
$ cat << EOF > ./credentials-velero
AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
AZURE_TENANT_ID=${AZURE_TENANT_ID}
AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_STORAGE_ACCOUNT_ACCESS_KEY=${AZURE_STORAGE_ACCOUNT_ACCES
```

```
S_KEY} 1
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

1 Obligatoire. Vous ne pouvez pas sauvegarder les images internes si le fichier **credentials-velero** ne contient que les informations d'identification du principal du service.

Vous utilisez le fichier **credentials-velero** pour créer un objet **Secret** pour Azure avant d'installer l'application de protection des données.

4.3.3.3. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.

IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

4.3.3.3.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

Le nom par défaut du site **Secret** est **cloud-credentials-azure**.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials-azure -n openshift-adp --from-file
cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

4.3.3.3.2. Création de secrets pour différentes informations d'identification

Si vos emplacements de sauvegarde et de snapshot utilisent des identifiants différents, vous devez créer deux objets **Secret**:

- Sauvegarde de l'emplacement **Secret** avec un nom personnalisé. Le nom personnalisé est spécifié dans le bloc **spec.backupLocations** de la ressource personnalisée (CR) **DataProtectionApplication**.

- Emplacement de l'instantané **Secret** avec le nom par défaut, **cloud-credentials-azure**. Cette adresse **Secret** n'est pas spécifiée dans la CR **DataProtectionApplication**.

Procédure

1. Créez un fichier **credentials-velero** pour l'emplacement de l'instantané dans le format approprié pour votre fournisseur de cloud.
2. Créez un site **Secret** pour l'emplacement de l'instantané avec le nom par défaut :

```
$ oc create secret generic cloud-credentials-azure -n openshift-adp --from-file
cloud=credentials-velero
```

3. Créez un fichier **credentials-velero** pour l'emplacement de sauvegarde dans le format approprié pour votre stockage d'objets.
4. Créez une adresse **Secret** pour l'emplacement de sauvegarde avec un nom personnalisé :

```
oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-
velero
```

5. Ajoutez le **Secret** avec le nom personnalisé au **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          storageAccount: <azure_storage_account_id>
          subscriptionId: <azure_subscription_id>
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: <custom_secret> 1
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
  snapshotLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

1 Emplacement de la sauvegarde **Secret** avec un nom personnalisé.

4.3.3.4. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

4.3.3.4.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Spécifier le sélecteur de nœud à fournir à Velero podSpec

4.3.3.4.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

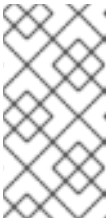
1. Indiquez la chaîne du certificat de l'autorité de certification codée en base64.
2. La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

4.3.3.5. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.
- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).
- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials-azure**.
- Si les emplacements de sauvegarde et de cliché utilisent des informations d'identification différentes, vous devez créer deux sites **Secrets**:
  - **Secret** avec un nom personnalisé pour l'emplacement de la sauvegarde. Vous ajoutez ce **Secret** au CR **DataProtectionApplication**.
  - **Secret** avec le nom par défaut, **cloud-credentials-azure**, pour l'emplacement de l'instantané. Ce site **Secret** n'est pas référencé dans le CR **DataProtectionApplication**.



NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - azure
        - openshift (1)
    restic:
      enable: true (2)
      podConfig:
        nodeSelector: <node selector> (3)
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> (4)
          storageAccount: <azure_storage_account_id> (5)
          subscriptionId: <azure_subscription_id> (6)
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure (7)
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name> (8)
          prefix: <prefix> (9)
  snapshotLocations: (10)
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

1. Le plugin **openshift** est obligatoire.
2. Définissez **false** si vous souhaitez désactiver l'installation de Restic. Restic déploie un ensemble de démons, ce qui signifie que chaque nœud de travailleur a des pods **Restic** en cours d'exécution. Vous configurez Restic pour les sauvegardes en ajoutant **spec.defaultVolumesToRestic: true** au CR **Backup**.
3. Spécifie le sélecteur de nœud à fournir à Restic podSpec.
4. Spécifiez le groupe de ressources Azure.
5. Indiquez l'identifiant du compte de stockage Azure.
6. Indiquez l'identifiant de l'abonnement Azure.
7. Si vous ne spécifiez pas cette valeur, le nom par défaut, **cloud-credentials-azure**, est utilisé. Si vous spécifiez un nom personnalisé, celui-ci est utilisé pour l'emplacement de la sauvegarde.
8. Spécifiez un bac comme emplacement de stockage des sauvegardes. Si le bac n'est pas un bac dédié aux sauvegardes Velero, vous devez spécifier un préfixe.
9. Spécifiez un préfixe pour les sauvegardes Velero, par exemple, **velero**, si le seau est utilisé à des fins multiples.
10. Il n'est pas nécessaire de spécifier un emplacement d'instantané si vous utilisez des instantanés CSI ou Restic pour sauvegarder des PV.

4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s
```

```
NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                     1         1         1       96s
```

4.3.3.5.1. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

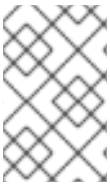
- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) Ajouter le plugin par défaut **csi**.

4.3.4. Installation et configuration de l'API OpenShift pour la protection des données avec Google Cloud Platform

Vous installez OpenShift API for Data Protection (OADP) avec Google Cloud Platform (GCP) en installant l'opérateur OADP. L'opérateur installe Velero 1.9.



NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Vous configurez GCP pour Velero, créez une adresse **Secret** par défaut, puis installez l'application de protection des données.

Pour installer l'OADP Operator dans un environnement réseau restreint, vous devez d'abord désactiver les sources OperatorHub par défaut et mettre en miroir le catalogue Operator. Voir Utilisation d'Operator Lifecycle Manager sur des réseaux restreints pour plus de détails.

4.3.4.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec des privilèges **cluster-admin**.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.
3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

4.3.4.2. Configuration de Google Cloud Platform

Vous configurez Google Cloud Platform (GCP) pour OpenShift API for Data Protection (OADP).

Conditions préalables

- Les outils **gcloud** et **gsutil** CLI doivent être installés. Voir la documentation de Google Cloud pour plus de détails.

Procédure

1. Connectez-vous à GCP :

```
$ gcloud auth login
```

2. Définir la variable **BUCKET**:

```
bUCKET=<bucket> 1
```

1 Indiquez le nom de votre seau.

3. Créez le seau de stockage :

```
$ gsutil mb gs://$BUCKET/
```

4. Définissez la variable **PROJECT_ID** sur votre projet actif :

```
$ PROJECT_ID=$(gcloud config get-value project)
```

5. Créer un compte de service :

```
$ gcloud iam service-accounts create velero \
    --display-name "Velero service account"
```

6. Dressez la liste de vos comptes de service :

```
$ gcloud iam service-accounts list
```

7. Définissez la variable **SERVICE_ACCOUNT_EMAIL** pour qu'elle corresponde à sa valeur **email**:

```
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list \
    --filter="displayName:Velero service account" \
    --format 'value(email)')
```

8. Attachez les politiques pour donner à l'utilisateur **velero** les permissions minimales nécessaires :

```
$ ROLE_PERMISSIONS=(
    compute.disks.get
    compute.disks.create
    compute.disks.createSnapshot
    compute.snapshots.get
    compute.snapshots.create
    compute.snapshots.useReadOnly
    compute.snapshots.delete
    compute.zones.get
)
```

9. Créez le rôle personnalisé **velero.server**:

```
$ gcloud iam roles create velero.server \
    --project $PROJECT_ID \
    --title "Velero Server" \
    --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"
```

10. Ajouter une politique IAM au projet :

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \
    --role projects/$PROJECT_ID/roles/velero.server
```

11. Mettre à jour le compte de service IAM :

```
$ gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

12. Enregistrer les clés du compte de service IAM dans le fichier **credentials-velero** dans le répertoire actuel :

```
$ gcloud iam service-accounts keys create credentials-velero \
    --iam-account $SERVICE_ACCOUNT_EMAIL
```

Vous utilisez le fichier **credentials-velero** pour créer un objet **Secret** pour GCP avant d'installer l'application de protection des données.

4.3.4.3. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

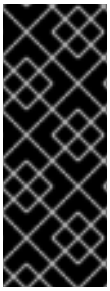
Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.



IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

4.3.4.3.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

Le nom par défaut du site **Secret** est **cloud-credentials-gcp**.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials-gcp -n openshift-adp --from-file
cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

4.3.4.3.2. Création de secrets pour différentes informations d'identification

Si vos emplacements de sauvegarde et de snapshot utilisent des identifiants différents, vous devez créer deux objets **Secret**:

- Sauvegarde de l'emplacement **Secret** avec un nom personnalisé. Le nom personnalisé est spécifié dans le bloc **spec.backupLocations** de la ressource personnalisée (CR) **DataProtectionApplication**.
- Emplacement de l'instantané **Secret** avec le nom par défaut, **cloud-credentials-gcp**. Cette adresse **Secret** n'est pas spécifiée dans la CR **DataProtectionApplication**.

Procédure

1. Créez un fichier **credentials-velero** pour l'emplacement de l'instantané dans le format approprié pour votre fournisseur de cloud.
2. Créez un site **Secret** pour l'emplacement de l'instantané avec le nom par défaut :

```
$ oc create secret generic cloud-credentials-gcp -n openshift-adp --from-file
cloud=credentials-velero
```

3. Créez un fichier **credentials-velero** pour l'emplacement de sauvegarde dans le format approprié pour votre stockage d'objets.

4. Créez une adresse **Secret** pour l'emplacement de sauvegarde avec un nom personnalisé :

```
oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-velero
```

5. Ajoutez le **Secret** avec le nom personnalisé au **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud
          name: <custom_secret> 1
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
  snapshotLocations:
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1
```

1 Emplacement de la sauvegarde **Secret** avec un nom personnalisé.

4.3.4.4. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

4.3.4.4.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Spécifier le sélecteur de nœud à fournir à Velero podSpec

4.3.4.4.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

1. Indiquez la chaîne du certificat de l'autorité de certification codée en base64.
2. La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

4.3.4.5. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.
- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).
- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials-gcp**.
- Si les emplacements de sauvegarde et de cliché utilisent des informations d'identification différentes, vous devez créer deux sites **Secrets**:
  - **Secret** avec un nom personnalisé pour l'emplacement de la sauvegarde. Vous ajoutez ce **Secret** au CR **DataProtectionApplication**.
  - **Secret** avec le nom par défaut, **cloud-credentials-gcp**, pour l'emplacement de l'instantané. Ce site **Secret** n'est pas référencé dans le CR **DataProtectionApplication**.

NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
```

```
spec:
  configuration:
    velero:
      defaultPlugins:
        - gcp
        - openshift (1)
    restic:
      enable: true (2)
      podConfig:
        nodeSelector: <node selector> (3)
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud
          name: cloud-credentials-gcp (4)
        objectStorage:
          bucket: <bucket_name> (5)
          prefix: <prefix> (6)
  snapshotLocations: (7)
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1 (8)
```

(1) Le plugin **openshift** est obligatoire.

(2) Définissez **false** si vous souhaitez désactiver l'installation de Restic. Restic déploie un ensemble de démons, ce qui signifie que chaque nœud de travailleur a des pods **Restic** en cours d'exécution. Vous configurez Restic pour les sauvegardes en ajoutant **spec.defaultVolumesToRestic: true** au CR **Backup**.

(3) Spécifie le sélecteur de nœud à fournir à Restic podSpec.

(4) Si vous ne spécifiez pas cette valeur, le nom par défaut, **cloud-credentials-gcp**, est utilisé. Si vous spécifiez un nom personnalisé, celui-ci est utilisé pour l'emplacement de la sauvegarde.

(5) Spécifiez un bac comme emplacement de stockage des sauvegardes. Si le bac n'est pas un bac dédié aux sauvegardes Velero, vous devez spécifier un préfixe.

(6) Spécifiez un préfixe pour les sauvegardes Velero, par exemple, **velero**, si le seau est utilisé à des fins multiples.

(7) Il n'est pas nécessaire de spécifier un emplacement d'instantané si vous utilisez des instantanés CSI ou Restic pour sauvegarder des PV.

(8) L'emplacement de l'instantané doit être situé dans la même région que les PV.

4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                       1         1         1       96s
```

4.3.4.5.1. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

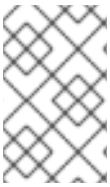
- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) Ajouter le plugin par défaut **csi**.

4.3.5. Installation et configuration de l'API OpenShift pour la protection des données avec Multicloud Object Gateway

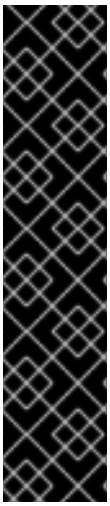
Vous installez OpenShift API for Data Protection (OADP) avec Multicloud Object Gateway (MCG) en installant l'opérateur OADP. L'opérateur installe Velero 1.9.



NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Vous configurez Multicloud Object Gateway en tant qu'emplacement de sauvegarde. MCG est un composant d'OpenShift Data Foundation. Vous configurez MCG comme emplacement de sauvegarde dans la ressource personnalisée (CR) **DataProtectionApplication**.



IMPORTANT

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

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

Vous créez un site **Secret** pour l'emplacement de sauvegarde, puis vous installez l'application de protection des données.

Pour installer l'OADP Operator dans un environnement réseau restreint, vous devez d'abord désactiver les sources OperatorHub par défaut et mettre en miroir le catalogue Operator. Pour plus de détails, voir Utilisation d'Operator Lifecycle Manager sur des réseaux restreints .

4.3.5.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec des privilèges **cluster-admin**.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.

3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

4.3.5.2. Récupération des informations d'identification de la passerelle d'objets multicloud

Vous devez récupérer les informations d'identification de Multicloud Object Gateway (MCG) afin de créer une ressource personnalisée (CR) **Secret** pour OpenShift API for Data Protection (OADP).

MCG est un composant d'OpenShift Data Foundation.

Conditions préalables

- Vous devez déployer OpenShift Data Foundation en utilisant le guide de déploiement OpenShift Data Foundation approprié.

Procédure

1. Obtenez le point de terminaison S3, **AWS_ACCESS_KEY_ID** et **AWS_SECRET_ACCESS_KEY** en exécutant la commande**describe** sur la ressource personnalisée **NooBaa**.
2. Créer un fichier **credentials-velero**:

```
$ cat << EOF > ./credentials-velero
[default]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
EOF
```

Vous utilisez le fichier **credentials-velero** pour créer un objet **Secret** lorsque vous installez l'application de protection des données.

4.3.5.3. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.

IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

4.3.5.3.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

Le nom par défaut du site **Secret** est **cloud-credentials**.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

4.3.5.3.2. Création de secrets pour différentes informations d'identification

Si vos emplacements de sauvegarde et de snapshot utilisent des identifiants différents, vous devez créer deux objets **Secret**:

- Sauvegarde de l'emplacement **Secret** avec un nom personnalisé. Le nom personnalisé est spécifié dans le bloc **spec.backupLocations** de la ressource personnalisée (CR) **DataProtectionApplication**.
- Emplacement de l'instantané **Secret** avec le nom par défaut, **cloud-credentials**. Cette adresse **Secret** n'est pas spécifiée dans la CR **DataProtectionApplication**.

Procédure

1. Créez un fichier **credentials-velero** pour l'emplacement de l'instantané dans le format approprié pour votre fournisseur de cloud.
2. Créez un site **Secret** pour l'emplacement de l'instantané avec le nom par défaut :

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

3. Créez un fichier **credentials-velero** pour l'emplacement de sauvegarde dans le format approprié pour votre stockage d'objets.
4. Créez une adresse **Secret** pour l'emplacement de sauvegarde avec un nom personnalisé :

```
oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-velero
```

5. Ajoutez le **Secret** avec le nom personnalisé au **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url>
          insecureSkipTLSVerify: "true"
```

```
      s3ForcePathStyle: "true"
    provider: aws
    default: true
    credential:
      key: cloud
      name:  <custom_secret> (1)
    objectStorage:
      bucket: <bucket_name>
      prefix: <prefix>
```

(1) Emplacement de la sauvegarde **Secret** avec un nom personnalisé.

4.3.5.4. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

4.3.5.4.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> (1)
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

(1) Spécifier le sélecteur de nœud à fournir à Velero podSpec

4.3.5.4.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

1. Indiquez la chaîne du certificat de l'autorité de certification codée en base64.
2. La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

4.3.5.5. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.
- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).

- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials**.
- Si les emplacements de sauvegarde et de cliché utilisent des informations d'identification différentes, vous devez créer deux sites **Secrets**:
  - **Secret** avec un nom personnalisé pour l'emplacement de la sauvegarde. Vous ajoutez ce **Secret** au CR **DataProtectionApplication**.
  - **Secret** avec le nom par défaut, **cloud-credentials**, pour l'emplacement de l'instantané. Ce site **Secret** n'est pas référencé dans le CR **DataProtectionApplication**.

NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift (1)
    restic:
      enable: true (2)
      podConfig:
        nodeSelector: <node selector> (3)
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url> (4)
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
        credential:
          key: cloud
          name: cloud-credentials (5)
```

```
    objectStorage:
      bucket: <bucket_name> 6
      prefix: <prefix> 7
```

1. Le plugin **openshift** est obligatoire.
2. Définissez **false** si vous souhaitez désactiver l'installation de Restic. Restic déploie un ensemble de démons, ce qui signifie que chaque nœud de travailleur a des pods **Restic** en cours d'exécution. Vous configurez Restic pour les sauvegardes en ajoutant **spec.defaultVolumesToRestic: true** au CR **Backup**.
3. Spécifie le sélecteur de nœud à fournir à Restic podSpec.
4. Spécifiez l'URL du point de terminaison S3.
5. Si vous ne spécifiez pas cette valeur, le nom par défaut, **cloud-credentials**, est utilisé. Si vous spécifiez un nom personnalisé, celui-ci est utilisé pour l'emplacement de la sauvegarde.
6. Spécifiez un bac comme emplacement de stockage des sauvegardes. Si le bac n'est pas un bac dédié aux sauvegardes Velero, vous devez spécifier un préfixe.
7. Spécifiez un préfixe pour les sauvegardes Velero, par exemple, **velero**, si le seau est utilisé à des fins multiples.

4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s
```

```
NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                     1         1         1       96s
```

4.3.5.5.1. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

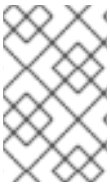
- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) Ajouter le plugin par défaut **csi**.

4.3.6. Installation et configuration de l'API OpenShift pour la protection des données avec OpenShift Data Foundation

Vous installez OpenShift API for Data Protection (OADP) avec OpenShift Data Foundation en installant l'opérateur OADP et en configurant un emplacement de sauvegarde et un emplacement d'instantané. Ensuite, vous installez l'application de protection des données.



NOTE

À partir d'OADP 1.0.4, toutes les versions d'OADP 1.0.*z* ne peuvent être utilisées qu'en tant que dépendance de l'opérateur MTC et ne sont pas disponibles en tant qu'opérateur autonome.

Vous pouvez configurer Multicloud Object Gateway ou tout stockage d'objets compatible S3 comme emplacement de sauvegarde.

IMPORTANT

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

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

Vous créez un site **Secret** pour l'emplacement de sauvegarde, puis vous installez l'application de protection des données.

Pour installer l'OADP Operator dans un environnement réseau restreint, vous devez d'abord désactiver les sources OperatorHub par défaut et mettre en miroir le catalogue Operator. Pour plus de détails, voir Utilisation d'Operator Lifecycle Manager sur des réseaux restreints .

4.3.6.1. Installation de l'opérateur OADP

Vous installez l'opérateur OpenShift API for Data Protection (OADP) sur OpenShift Container Platform 4.12 en utilisant Operator Lifecycle Manager (OLM).

L'opérateur OADP installe Velero 1.9.

Conditions préalables

- Vous devez être connecté en tant qu'utilisateur avec des privilèges **cluster-admin**.

Procédure

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Utilisez le champ **Filter by keyword** pour trouver le **OADP Operator**.
3. Sélectionnez le site **OADP Operator** et cliquez sur **Install**.
4. Cliquez sur **Install** pour installer l'opérateur dans le projet **openshift-adp**.
5. Cliquez sur **Operators → Installed Operators** pour vérifier l'installation.

4.3.6.2. A propos des emplacements de sauvegarde et d'instantané et de leurs secrets

Vous spécifiez les emplacements de sauvegarde et d'instantané ainsi que leurs secrets dans la ressource personnalisée (CR) **DataProtectionApplication**.

Emplacements de sauvegarde
Vous spécifiez un stockage d'objets compatible S3, tel que Multicloud Object Gateway, Noobaa ou Minio, en tant qu'emplacement de sauvegarde.

Velero sauvegarde les ressources d'OpenShift Container Platform, les objets Kubernetes et les images internes en tant que fichier d'archive sur le stockage d'objets.

Lieux de l'instantané
Si vous utilisez l'API d'instantané native de votre fournisseur de cloud computing pour sauvegarder des volumes persistants, vous devez spécifier le fournisseur de cloud computing comme emplacement d'instantané.

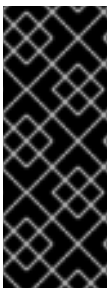
Si vous utilisez des instantanés de l'interface de stockage de conteneurs (CSI), vous n'avez pas besoin de spécifier un emplacement d'instantané car vous créerez un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.

Si vous utilisez Restic, vous n'avez pas besoin de spécifier un emplacement d'instantané car Restic sauvegarde le système de fichiers sur le stockage objet.

Secrets
Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement d'instantané, vous créez un emplacement par défaut **Secret**.

Si les emplacements de sauvegarde et d'instantané utilisent des informations d'identification différentes, vous créez deux objets secrets :

- **Secret** personnalisé pour l'emplacement de sauvegarde, que vous spécifiez dans le CR **DataProtectionApplication**.
- **Secret** par défaut pour l'emplacement de l'instantané, qui n'est pas référencé dans le CR **DataProtectionApplication**.



IMPORTANT

L'application de protection des données nécessite une adresse par défaut **Secret**. Dans le cas contraire, l'installation échouera.

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer un site **Secret** par défaut avec un fichier **credentials-velero** vide.

4.3.6.2.1. Création d'un secret par défaut

Vous créez un site **Secret** par défaut si vos emplacements de sauvegarde et de cliché utilisent les mêmes informations d'identification ou si vous n'avez pas besoin d'un emplacement de cliché.

NOTE

La ressource personnalisée (CR) **DataProtectionApplication** nécessite une ressource par défaut **Secret**. Sinon, l'installation échouera. Si le nom de l'emplacement de sauvegarde **Secret** n'est pas spécifié, le nom par défaut est utilisé.

Si vous ne souhaitez pas utiliser les informations d'identification de l'emplacement de sauvegarde lors de l'installation, vous pouvez créer un site **Secret** avec le nom par défaut en utilisant un fichier **credentials-velero** vide.

Conditions préalables

- Votre stockage d'objets et votre stockage en nuage, le cas échéant, doivent utiliser les mêmes informations d'identification.
- Vous devez configurer le stockage d'objets pour Velero.
- Vous devez créer un fichier **credentials-velero** pour le stockage d'objets dans le format approprié.

Procédure

- Créez un site **Secret** avec le nom par défaut :

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

Le site **Secret** est référencé dans le bloc **spec.backupLocations.credential** de la CR **DataProtectionApplication** lorsque vous installez l'application de protection des données.

4.3.6.3. Configuration de l'application de protection des données

Vous pouvez configurer l'application de protection des données en définissant les allocations de ressources Velero ou en activant les certificats CA auto-signés.

4.3.6.3.1. Paramétrage de l'allocation des ressources CPU et mémoire de Velero

Vous définissez les allocations de ressources CPU et mémoire pour le pod **Velero** en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifiez les valeurs dans le bloc **spec.configuration.velero.podConfig.ResourceAllocations** du manifeste **DataProtectionApplication** CR, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Spécifier le sélecteur de nœud à fournir à Velero podSpec

4.3.6.3.2. Activation des certificats CA auto-signés

Vous devez activer un certificat CA auto-signé pour le stockage d'objets en modifiant le manifeste de ressources personnalisées (CR) **DataProtectionApplication** afin d'éviter une erreur **certificate signed by unknown authority**.

Conditions préalables

- L'opérateur OpenShift API for Data Protection (OADP) doit être installé.

Procédure

- Modifier les paramètres **spec.backupLocations.velero.objectStorage.caCert** et **spec.backupLocations.velero.config** du manifeste **DataProtectionApplication** CR :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
...
```

1 Indiquez la chaîne du certificat de l'autorité de certification codée en base64.

2 La configuration **insecureSkipTLSVerify** peut être réglée sur **"true"** ou **"false"**. Si elle est réglée sur **"true"**, la sécurité SSL/TLS est désactivée. Si la configuration est **"false"**, la sécurité SSL/TLS est activée.

4.3.6.4. Installation de l'application de protection des données

Vous installez l'application de protection des données (DPA) en créant une instance de l'API **DataProtectionApplication**.

Conditions préalables

- Vous devez installer l'opérateur OADP.
- Vous devez configurer le stockage d'objets comme emplacement de sauvegarde.

- Si vous utilisez des instantanés pour sauvegarder des PV, votre fournisseur de cloud computing doit prendre en charge une API d'instantanés native ou des instantanés de l'interface de stockage de conteneurs (CSI).
- Si les emplacements de sauvegarde et d'instantané utilisent les mêmes informations d'identification, vous devez créer un site **Secret** avec le nom par défaut, **cloud-credentials**.

NOTE

Si vous ne souhaitez pas spécifier d'emplacements de sauvegarde ou d'instantanés lors de l'installation, vous pouvez créer une adresse **Secret** par défaut avec un fichier **credentials-velero** vide. S'il n'y a pas de **Secret** par défaut, l'installation échouera.

Procédure

1. Cliquez sur **Operators → Installed Operators** et sélectionnez l'opérateur OADP.
2. Sous **Provided APIs**, cliquez sur **Create instance** dans la boîte **DataProtectionApplication**.
3. Cliquez sur **YAML View** et mettez à jour les paramètres du manifeste **DataProtectionApplication**:
4. Cliquez sur **Create**.
5. Vérifiez l'installation en consultant les ressources de l'OADP :

```
$ oc get all -n openshift-adp
```

Exemple de sortie

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                    TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                          1         1         1       96s
```

4.3.6.4.1. Configurer NooBaa pour la reprise après sinistre sur OpenShift Data Foundation

Si vous utilisez le stockage en cluster pour votre bucket NooBaa **backupStorageLocation** sur OpenShift Data Foundation, configurez NooBaa en tant que magasin d'objets externe.

AVERTISSEMENT

Si NooBaa n'est pas configuré comme un magasin d'objets externe, les sauvegardes risquent de ne pas être disponibles.

Procédure

- Configurez NooBaa en tant que magasin d'objets externe comme décrit dans Ajouter des ressources de stockage pour l'hybride ou le Multicloud.

4.3.6.4.2. Activation de l'ISC dans l'application de protection des données CR

Vous activez l'interface de stockage de conteneurs (CSI) dans la ressource personnalisée (CR) **DataProtectionApplication** afin de sauvegarder des volumes persistants à l'aide d'instantanés CSI.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.

Procédure

- Modifiez le CR **DataProtectionApplication**, comme dans l'exemple suivant :

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) Ajouter le plugin par défaut **csi**.

4.3.7. Désinstallation de l'API OpenShift pour la protection des données

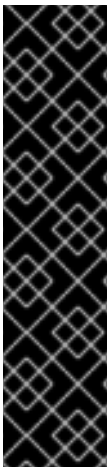
Vous désinstallez OpenShift API for Data Protection (OADP) en supprimant l'opérateur OADP. Pour plus d'informations, reportez-vous à la section Suppression des opérateurs d'un cluster .

4.4. SAUVEGARDE ET RESTAURATION

4.4.1. Sauvegarde des applications

Vous sauvegardez les applications en créant une **Backup** ressource personnalisée (CR).

Le CR **Backup** crée des fichiers de sauvegarde pour les ressources Kubernetes et les images internes, sur le stockage d'objets S3, et des instantanés pour les volumes persistants (PV), si le fournisseur de cloud utilise une API d'instantané native ou l'interface de stockage de conteneurs (CSI) pour créer des instantanés, comme OpenShift Data Foundation 4. Pour plus d'informations, voir Instantanés de volume CSI.



IMPORTANT

L'API **CloudStorage** pour le stockage S3 est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

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

Si votre fournisseur de cloud dispose d'une API d'instantané native ou prend en charge les instantanés de l'interface de stockage de conteneurs (CSI), le CR **Backup** sauvegarde les volumes persistants en créant des instantanés. Pour plus d'informations, voir Overview of CSI volume snapshots dans la documentation OpenShift Container Platform.

Si votre fournisseur de cloud computing ne prend pas en charge les instantanés ou si vos applications se trouvent sur des volumes de données NFS, vous pouvez créer des sauvegardes à l'aide de Restic.

Vous pouvez créer des crochets de sauvegarde pour exécuter des commandes avant ou après l'opération de sauvegarde.

Vous pouvez planifier des sauvegardes en créant un CR**Schedule** au lieu d'un CR **Backup**.

4.4.1.1. Création d'un CR de sauvegarde

Vous sauvegardez les images Kubernetes, les images internes et les volumes persistants (PV) en créant une ressource personnalisée (CR) **Backup**.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.
- Conditions préalables relatives à l'emplacement de la sauvegarde :
  - Le stockage d'objets S3 doit être configuré pour Velero.
  - Un emplacement de sauvegarde doit être configuré dans le CR **DataProtectionApplication**.
- Conditions préalables pour l'emplacement des instantanés :

- Votre fournisseur de cloud computing doit disposer d'une API d'instantané native ou prendre en charge les instantanés de l'interface de stockage de conteneurs (CSI).
- Pour les instantanés CSI, vous devez créer un CR **VolumeSnapshotClass** pour enregistrer le pilote CSI.
- Un emplacement de volume doit être configuré dans le CR **DataProtectionApplication**.

Procédure

1. Récupérez les CR **backupStorageLocations** en entrant la commande suivante :

```
$ oc get backupStorageLocations -n openshift-adp
```

Exemple de sortie

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. Créez un CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  hooks: {}
  includedNamespaces:
  - <namespace> 1
  includedResources: [] 2
  excludedResources: [] 3
  storageLocation: <velero-sample-1> 4
  ttl: 720h0m0s
  labelSelector: 5
  - matchLabels:
      app=<label_1>
  - matchLabels:
      app=<label_2>
  - matchLabels:
      app=<label_3>
  orlabelSelectors: 6
  - matchLabels:
      app=<label_1>
  - matchLabels:
      app=<label_2>
  - matchLabels:
      app=<label_3>
```

1 Spécifier un tableau d'espaces de noms à sauvegarder.

2

Facultatif : Spécifiez un tableau de ressources à inclure dans la sauvegarde. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement

3 Facultatif : Spécifiez un tableau de ressources à exclure de la sauvegarde. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement qualifiées.

4 Indiquez le nom du CR **backupStorageLocations**.

5 Sauvegarde des ressources qui ont toutes les étiquettes spécifiées.

6 Sauvegarde des ressources qui ont une ou plusieurs des étiquettes spécifiées.

3. Vérifiez que l'état de la CR **Backup** est **Completed**:

```
$ oc get backup -n openshift-adp <backup> -o jsonpath='{.status.phase}'
```

4.4.1.2. Sauvegarde de volumes persistants avec des instantanés CSI

Vous sauvegardez des volumes persistants avec des instantanés de l'interface de stockage de conteneurs (CSI) en modifiant la ressource personnalisée (CR) **VolumeSnapshotClass** du stockage en nuage avant de créer la CR **Backup**.

Conditions préalables

- Le fournisseur de services en nuage doit prendre en charge les instantanés CSI.
- Vous devez activer le CSI sur le site **DataProtectionApplication** CR.

Procédure

- Ajouter la paire clé-valeur **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** à la CR **VolumeSnapshotClass**:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true"
driver: <csi_driver>
deletionPolicy: Retain
```

Vous pouvez maintenant créer un CR **Backup**.

4.4.1.3. Sauvegarde des applications avec Restic

Vous sauvegardez les ressources Kubernetes, les images internes et les volumes persistants avec Restic en modifiant la ressource personnalisée (CR) **Backup**.

Il n'est pas nécessaire de spécifier un emplacement d'instantané dans le CR **DataProtectionApplication**.

IMPORTANT

Restic ne prend pas en charge la sauvegarde des volumes **hostPath**. Pour plus d'informations, voir les limitations supplémentaires de Restic.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Vous ne devez pas désactiver l'installation par défaut de Restic en remplaçant **spec.configuration.restic.enable** par **false** dans le CR **DataProtectionApplication**.
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.

Procédure

- Modifiez le CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  defaultVolumesToRestic: true (1)
...
```

(1) Ajouter **defaultVolumesToRestic: true** au bloc **spec**.

4.4.1.4. Utilisation de Data Mover pour les instantanés CSI

IMPORTANT

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

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

Le Data Mover de l'OADP 1.1.2 permet aux clients de sauvegarder les instantanés de volume de l'interface de stockage de conteneurs (CSI) dans un magasin d'objets distant. Lorsque le Data Mover est activé, vous pouvez restaurer des applications avec état à partir du magasin en cas de panne, de suppression accidentelle ou de corruption du cluster. La solution OADP 1.1.2 Data Mover utilise l'option Restic de VolSync.

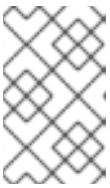
NOTE

Data Mover ne prend en charge que la sauvegarde et la restauration des instantanés de volumes CSI.

Actuellement, Data Mover ne prend pas en charge les buckets de Google Cloud Storage (GCS).

Conditions préalables

- Vous avez vérifié que les ressources personnalisées (CR) **StorageClass** et **VolumeSnapshotClass** prennent en charge CSI.
- Vous avez vérifié qu'un seul CR **volumeSnapshotClass** porte l'annotation **snapshot.storage.kubernetes.io/is-default-class: true**.
- Vous avez vérifié qu'un seul CR **storageClass** porte l'annotation **storageclass.kubernetes.io/is-default-class: true**.
- Vous avez inclus le label **velero.io/csi-volumesnapshot-class: 'true'** dans votre CR **VolumeSnapshotClass**.
- Vous avez installé l'opérateur VolSync en utilisant le gestionnaire du cycle de vie de l'opérateur (OLM).



NOTE

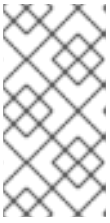
L'opérateur VolSync n'est nécessaire que pour l'utilisation du Data Mover de l'aperçu technologique. Il n'est pas nécessaire pour utiliser les fonctions de production de l'OADP.

- Vous avez installé l'opérateur OADP en utilisant OLM.

Procédure

1. Configurez un secret Restic en créant un fichier **.yaml** comme suit :

```
apiVersion: v1
kind: Secret
metadata:
  name: <secret_name>
  namespace: openshift-adp
type: Opaque
stringData:
  RESTIC_PASSWORD: <secure_restic_password>
```



NOTE

Par défaut, l'opérateur recherche un secret nommé **dm-credential**. Si vous utilisez un nom différent, vous devez le spécifier dans une application de protection des données (DPA) CR à l'aide de **dpa.spec.features.dataMover.credentialName**.

2. Créez un DPA CR similaire à l'exemple suivant. Les plugins par défaut incluent CSI.

Exemple de demande de protection des données (DPA) CR

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  features:
    dataMover:
      enable: true
      credentialName: <secret_name> 1
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <bucket_prefix>
        provider: aws
  configuration:
    restic:
      enable: <true_or_false>
    velero:
      defaultPlugins:
        - openshift
        - aws
        - csi
```

1 Ajoutez le nom secret Restic de l'étape précédente. Si cela n'est pas fait, le nom secret par défaut **dm-credential** est utilisé.

L'opérateur OADP installe deux définitions de ressources personnalisées (CRD), **VolumeSnapshotBackup** et **VolumeSnapshotRestore**.

Exemple VolumeSnapshotBackup CRD

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotBackup
metadata:
  name: <vsb_name>
  namespace: <namespace_name> 1
spec:
  volumeSnapshotContent:
    name: <snapcontent_name>
  protectedNamespace: <adp_namespace>
  resticSecretRef:
    name: <restic_secret_name>
```

1 Spécifiez l'espace de noms dans lequel le cliché instantané du volume existe.

Exemple VolumeSnapshotRestore CRD

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotRestore
metadata:
  name: <vsr_name>
  namespace: <namespace_name> 1
spec:
  protectedNamespace: <protected_ns> 2
  resticSecretRef:
    name: <restic_secret_name>
  volumeSnapshotMoverBackupRef:
    sourcePVCData:
      name: <source_pvc_name>
      size: <source_pvc_size>
    resticrepository: <your_restic_repo>
    volumeSnapshotClassName: <vsclass_name>
```

1 Spécifiez l'espace de noms dans lequel le cliché instantané du volume existe.

2 Indiquez l'espace de noms dans lequel l'opérateur est installé. La valeur par défaut est **openshift-adp**.

3. Vous pouvez sauvegarder un instantané de volume en procédant comme suit :

   a. Créer un CR de sauvegarde :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup_name>
  namespace: <protected_ns> 1
spec:
  includedNamespaces:
  - <app_ns>
  storageLocation: velero-sample-1
```

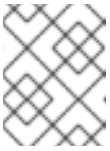
1 Indiquez l'espace de noms dans lequel l'opérateur est installé. L'espace de noms par défaut est **openshift-adp**.

   b. Attendez jusqu'à 10 minutes et vérifiez si l'état de **VolumeSnapshotBackup** CR est **Completed** en entrant les commandes suivantes :

```
$ oc get vsb -n <app_ns>
```

```
$ oc get vsb <vsb_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

Un instantané est créé dans le magasin d'objets configuré dans le DPA.



NOTE

Si l'état de **VolumeSnapshotBackup** CR devient **Failed**, reportez-vous aux registres Velero pour le dépannage.

4. Vous pouvez restaurer un instantané de volume en procédant comme suit :

   a. Supprimez l'espace de noms de l'application et le site **volumeSnapshotContent** qui a été créé par le plugin Velero CSI.

   b. Créez un CR **Restore** et définissez **restorePVs** comme **true**.

   Exemple Restore CR

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
  restorePVs: true
```

   c. Attendez jusqu'à 10 minutes et vérifiez si l'état de **VolumeSnapshotRestore** CR est **Completed** en entrant la commande suivante :

```
$ oc get vsr -n <app_ns>
```

```
$ oc get vsr <vsr_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

   d. Vérifiez si les données et les ressources de votre application ont été restaurées.

NOTE

Si l'état de **VolumeSnapshotRestore** CR devient "Échec", reportez-vous aux journaux Velero pour le dépannage.

Ressources complémentaires

- Installation des opérateurs sur les clusters pour les administrateurs
- Installation d'opérateurs dans des espaces de noms pour les non-administrateurs

4.4.1.5. Création de crochets de sauvegarde

Vous créez des crochets de sauvegarde pour exécuter des commandes dans un conteneur d'un pod en modifiant la ressource personnalisée (CR) **Backup**.

Pre s'exécutent avant la sauvegarde du pod. *Post* s'exécutent après la sauvegarde.

Procédure

- Ajoutez un crochet au bloc **spec.hooks** du CR **Backup**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces: 2
        - <namespace>
        includedResources: []
        - pods 3
        excludedResources: [] 4
        labelSelector: 5
          matchLabels:
            app: velero
            component: server
        pre: 6
          - exec:
              container: <container> 7
              command:
              - /bin/uname 8
              - -a
              onError: Fail 9
              timeout: 30s 10
        post: 11
...
```

1. Facultatif : vous pouvez spécifier les espaces de noms auxquels le crochet s'applique. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.
2. Facultatif : vous pouvez spécifier des espaces de noms auxquels le crochet ne s'applique pas.
3. Actuellement, les pods sont la seule ressource prise en charge à laquelle les crochets peuvent s'appliquer.
4. Facultatif : vous pouvez spécifier les ressources auxquelles le crochet ne s'applique pas.
5. Facultatif : Ce crochet ne s'applique qu'aux objets correspondant à l'étiquette. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.
6. Tableau de crochets à exécuter avant la sauvegarde.
7. Facultatif : si le conteneur n'est pas spécifié, la commande s'exécute dans le premier conteneur du pod.
8. Il s'agit du point d'entrée du conteneur init ajouté.
9. Les valeurs autorisées pour le traitement des erreurs sont **Fail** et **Continue**. La valeur par défaut est **Fail**.

10 Facultatif : durée d'attente pour l'exécution des commandes. La valeur par défaut est **30s**.

11 Ce bloc définit un tableau de hooks à exécuter après la sauvegarde, avec les mêmes paramètres que les hooks de pré-sauvegarde.

4.4.1.6. Planification des sauvegardes

Vous planifiez les sauvegardes en créant une ressource personnalisée (CR) **Schedule** au lieu d'une CR **Backup**.

AVERTISSEMENT

Laissez suffisamment de temps dans votre calendrier de sauvegarde pour qu'une sauvegarde se termine avant qu'une autre ne soit créée.

Par exemple, si la sauvegarde d'un espace de noms prend généralement 10 minutes, ne planifiez pas de sauvegardes plus fréquentes que toutes les 15 minutes.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.

Procédure

1. Récupérer les CR de **backupStorageLocations**:

```
$ oc get backupStorageLocations -n openshift-adp
```

Exemple de sortie

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. Créez un CR **Schedule**, comme dans l'exemple suivant :

```
$ cat << EOF | oc apply -f -
apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: <schedule>
  namespace: openshift-adp
spec:
  schedule: 0 7 * * * 1
  template:
    hooks: {}
    includedNamespaces:
```

```
    - <namespace> 2
    storageLocation: <velero-sample-1> 3
    defaultVolumesToRestic: true 4
    ttl: 720h0m0s
EOF
```

1. **cron** expression to schedule the backup, for example, **0 7 * * *** to perform a backup every day at 7:00.
2. Tableau des espaces de noms à sauvegarder.
3. Nom du CR **backupStorageLocations**.
4. Facultatif : Ajoutez la paire clé-valeur **defaultVolumesToRestic: true** si vous sauvegardez des volumes avec Restic.

3. Vérifiez que l'état de **Schedule** CR est **Completed** après l'exécution de la sauvegarde programmée :

```
$ oc get schedule -n openshift-adp <schedule> -o jsonpath='{.status.phase}'
```

4.4.1.7. Suppression des sauvegardes

Vous pouvez supprimer les fichiers de sauvegarde en supprimant la ressource personnalisée (CR) **Backup**.

> **AVERTISSEMENT**
>
> Une fois que vous avez supprimé le CR **Backup** et les données de stockage d'objets associées, vous ne pouvez pas récupérer les données supprimées.

Conditions préalables

- Vous avez créé un CR **Backup**.
- Vous connaissez le nom du CR **Backup** et l'espace de noms qui le contient.
- Vous avez téléchargé l'outil Velero CLI.
- Vous pouvez accéder au binaire Velero dans votre cluster.

Procédure

- Choisissez l'une des actions suivantes pour supprimer le CR **Backup**:
  - Pour supprimer le CR **Backup** et conserver les données de stockage de l'objet associé, exécutez la commande suivante :

```
oc delete backup <backup_CR_name> -n <velero_namespace>
```

- Pour supprimer le CR **Backup** et les données de stockage d'objets associées, exécutez la commande suivante :

```
$ velero backup delete <backup_CR_name> -n <velero_namespace>
```

Où ?

<backup_CR_name>
Spécifie le nom de la ressource personnalisée **Backup**.

<velero_namespace>
Spécifie l'espace de noms qui contient la ressource personnalisée **Backup**.

Ressources complémentaires

- Télécharger l'outil Velero CLI

4.4.2. Restauration des applications

Vous restaurez les sauvegardes de l'application en créant une ressource personnalisée (CR) à l'adresse**Restore** .

Vous pouvez créer des crochets de restauration pour exécuter des commandes dans les conteneurs d'initialisation, avant le démarrage du conteneur d'application ou dans le conteneur d'application lui-même.

4.4.2.1. Création d'un CR de restauration

Vous restaurez une ressource personnalisée (CR) **Backup** en créant une CR **Restore**.

Conditions préalables

- Vous devez installer l'opérateur OpenShift API for Data Protection (OADP).
- Le CR **DataProtectionApplication** doit être dans un état **Ready**.
- Vous devez avoir un Velero **Backup** CR.
- Ajustez la taille demandée pour que la capacité du volume persistant (PV) corresponde à la taille demandée au moment de la sauvegarde.

Procédure

1. Créez un CR **Restore**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  backupName: <backup> 1
  includedResources: [] 2
  excludedResources:
```

```
- nodes
- events
- events.events.k8s.io
- backups.velero.io
- restores.velero.io
- resticrepositories.velero.io
restorePVs: true
```

1. Nom du CR **Backup**.
2. Facultatif. Spécifiez un tableau de ressources à inclure dans le processus de restauration. Les ressources peuvent être des raccourcis (par exemple, "po" pour "pods") ou être entièrement qualifiées. Si rien n'est spécifié, toutes les ressources sont incluses.

2. Vérifiez que l'état du CR **Restore** est **Completed** en entrant la commande suivante :

```
$ oc get restore -n openshift-adp <restore> -o jsonpath='{.status.phase}'
```

3. Vérifiez que les ressources de sauvegarde ont été restaurées en entrant la commande suivante :

```
$ oc get all -n <namespace> 1
```

1. Namespace que vous avez sauvegardé.

4. Si vous utilisez Restic pour restaurer les objets **DeploymentConfig** ou si vous utilisez des crochets post-restauration, exécutez le script de nettoyage **dc-restic-post-restore.sh** en entrant la commande suivante :

```
bash dc-restic-post-restore.sh <restore-name>
```

NOTE

Au cours du processus de restauration, les plug-ins OADP Velero réduisent les objets **DeploymentConfig** et restaurent les pods en tant que pods autonomes pour éviter que le cluster ne supprime les pods **DeploymentConfig** restaurés immédiatement après la restauration et pour permettre aux hooks Restic et post-restauration de terminer leurs actions sur les pods restaurés. Le script de nettoyage supprime ces pods déconnectés et met à l'échelle tous les objets **DeploymentConfig** jusqu'au nombre approprié de répliques.

Exemple 4.1. dc-restic-post-restore.sh script de nettoyage

```
#!/bin/bash
set -e

# if sha256sum exists, use it to check the integrity of the file
if command -v sha256sum >/dev/null 2>&1; then
  CHECKSUM_CMD="sha256sum"
else
  CHECKSUM_CMD="shasum -a 256"
fi
```

```
label_name () {
    if [ "${#1}" -le "63" ]; then
 echo $1
 return
    fi
    sha=$(echo -n $1|$CHECKSUM_CMD)
    echo "${1:0:57}${sha:0:6}"
}

OADP_NAMESPACE=${OADP_NAMESPACE:=openshift-adp}

if [[ $# -ne 1 ]]; then
    echo "usage: ${BASH_SOURCE} restore-name"
    exit 1
fi

echo using OADP Namespace $OADP_NAMESPACE
echo restore: $1

label=$(label_name $1)
echo label: $label

echo Deleting disconnected restore pods
oc delete pods -l oadp.openshift.io/disconnected-from-dc=$label

for dc in $(oc get dc --all-namespaces -l oadp.openshift.io/replicas-modified=$label -o
jsonpath='{range .items[*]}{.metadata.namespace}{","}{.metadata.name}{","}
{.metadata.annotations.oadp\.openshift\.io/original-replicas}{","}
{.metadata.annotations.oadp\.openshift\.io/original-paused}{"\n"}')
do
    IFS=',' read -ra dc_arr <<< "$dc"
    if [ ${#dc_arr[0]} -gt 0 ]; then
 echo Found deployment ${dc_arr[0]}/${dc_arr[1]}, setting replicas: ${dc_arr[2]}, paused:
${dc_arr[3]}
 cat <<EOF | oc patch dc  -n ${dc_arr[0]} ${dc_arr[1]} --patch-file /dev/stdin
spec:
  replicas: ${dc_arr[2]}
  paused: ${dc_arr[3]}
EOF
    fi
done
```

4.4.2.2. Création de crochets de restauration

Vous créez des crochets de restauration pour exécuter des commandes dans un conteneur dans un pod tout en restaurant votre application en modifiant la ressource personnalisée (CR) **Restore**.

Vous pouvez créer deux types de crochets de restauration :

- Un crochet **init** ajoute un conteneur init à un pod pour effectuer des tâches de configuration avant que le conteneur d'application ne démarre.
  Si vous restaurez une sauvegarde Restic, le conteneur init **restic-wait** est ajouté avant le conteneur init restore hook.

- Un hook **exec** exécute des commandes ou des scripts dans un conteneur d'un pod restauré.

Procédure

- Ajoutez un crochet au bloc **spec.hooks** du CR **Restore**, comme dans l'exemple suivant :

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces:
        - <namespace>
        includedResources:
        - pods 2
        excludedResources: []
        labelSelector: 3
          matchLabels:
            app: velero
            component: server
        postHooks:
        - init:
            initContainers:
            - name: restore-hook-init
              image: alpine:latest
              volumeMounts:
              - mountPath: /restores/pvc1-vm
                name: pvc1-vm
              command:
              - /bin/ash
              - -c
            timeout: 4
        - exec:
            container: <container> 5
            command:
            - /bin/bash 6
            - -c
            - "psql < /backup/backup.sql"
            waitTimeout: 5m 7
            execTimeout: 1m 8
            onError: Continue 9
```

1. Facultatif : Tableau des espaces de noms auxquels le crochet s'applique. Si cette valeur n'est pas spécifiée, le crochet s'applique à tous les espaces de noms.

2. Actuellement, les pods sont la seule ressource prise en charge à laquelle les crochets peuvent s'appliquer.

3. Facultatif : Ce crochet ne s'applique qu'aux objets correspondant au sélecteur d'étiquette.
4. Facultatif : Timeout indique la durée maximale pendant laquelle Velero attend la fin de **initContainers**.
5. Facultatif : si le conteneur n'est pas spécifié, la commande s'exécute dans le premier conteneur du pod.
6. Il s'agit du point d'entrée du conteneur init ajouté.
7. Facultatif : durée d'attente pour qu'un conteneur soit prêt. Cette durée doit être suffisante pour que le conteneur démarre et que tous les crochets précédents dans le même conteneur soient terminés. S'il n'est pas défini, le processus de restauration attend indéfiniment.
8. Facultatif : durée d'attente pour l'exécution des commandes. La valeur par défaut est **30s**.
9. Les valeurs autorisées pour le traitement des erreurs sont **Fail** et **Continue**:
    - **Continue**: Seuls les échecs de commande sont consignés.
    - **Fail**: Plus aucun crochet de restauration n'est exécuté dans aucun conteneur, dans aucun pod. Le statut du CR **Restore** sera **PartiallyFailed**.

4.5. DÉPANNAGE

Vous pouvez déboguer les ressources personnalisées (CR) Velero en utilisant l'outil CLI d'OpenShift ou l'outil CLI de Velero. L'outil Velero CLI fournit des journaux et des informations plus détaillés.

Vous pouvez vérifier les problèmes d'installation, de sauvegarde et de restauration de la CR , ainsi que les problèmes liés à Restic.

Vous pouvez collecter des journaux, des informations CR et des données métriques Prometheus à l'aide de l'outil**must-gather** .

Vous pouvez obtenir l'outil Velero CLI par :

- Télécharger l'outil Velero CLI
- Accès au binaire Velero dans le déploiement Velero dans le cluster

4.5.1. Télécharger l'outil Velero CLI

Vous pouvez télécharger et installer l'outil Velero CLI en suivant les instructions de la page de documentation Velero.

La page comprend des instructions pour :

- macOS en utilisant Homebrew
- GitHub
- Windows en utilisant Chocolatey

Conditions préalables

- Vous avez accès à un cluster Kubernetes, v1.16 ou plus récent, avec le DNS et la mise en réseau des conteneurs activés.
- Vous avez installé **kubectl** localement.

Procédure

1. Ouvrez un navigateur et accédez à la page "Installer le CLI" sur le site web de Velero .
2. Suivez la procédure appropriée pour macOS, GitHub ou Windows.
3. Téléchargez la version de Velero correspondant à votre version d'OADP et d'OpenShift Container Platform selon le tableau suivant :

Tableau 4.2. OADP-Velero-OpenShift Container Platform relation de version

Version de l'OADP	Version Velero	Version d'OpenShift Container Platform
1.0.0	1.7	4.6 et plus
1.0.1	1.7	4.6 et plus
1.0.2	1.7	4.6 et plus
1.0.3	1.7	4.6 et plus
1.1.0	1.9	4.9 et plus
1.1.1	1.9	4.9 et plus
1.1.2	1.9	4.9 et plus

4.5.2. Accès au binaire Velero dans le déploiement Velero dans le cluster

Vous pouvez utiliser une commande shell pour accéder au binaire Velero dans le déploiement Velero dans le cluster.

Conditions préalables

- Votre ressource personnalisée **DataProtectionApplication** a le statut **Reconcile complete**.

Procédure

- Entrez la commande suivante pour définir l'alias nécessaire :

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

4.5.3. Déboguer les ressources Velero avec l'outil OpenShift CLI

Vous pouvez déboguer un échec de sauvegarde ou de restauration en vérifiant les ressources personnalisées Velero (CRs) et le journal du pod **Velero** avec l'outil CLI d'OpenShift.

Velero CRs
Utilisez la commande **oc describe** pour obtenir un résumé des avertissements et des erreurs associés à un CR **Backup** ou **Restore**:

```
oc describe <velero_cr> <cr_name>
```

Billes de pods Velero
Utilisez la commande **oc logs** pour récupérer les journaux du pod **Velero**:

```
$ oc logs pod/<velero>
```

Journaux de débogage du pod Velero
Vous pouvez spécifier le niveau de journalisation de Velero dans la ressource **DataProtectionApplication** comme le montre l'exemple suivant.

NOTE

Cette option est disponible à partir de l'OADP 1.0.3.

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
spec:
  configuration:
    velero:
      logLevel: warning
```

Les valeurs suivantes sont disponibles sur **logLevel**:

- **trace**
- **debug**
- **info**
- **warning**
- **error**
- **fatal**
- **panic**

Il est recommandé d'utiliser **debug** pour la plupart des journaux.

4.5.4. Déboguer les ressources Velero avec l'outil Velero CLI

Vous pouvez déboguer **Backup** et **Restore** custom resources (CRs) et récupérer les journaux avec l'outil Velero CLI.

L'outil Velero CLI fournit des informations plus détaillées que l'outil OpenShift CLI.

Syntaxe
Utilisez la commande **oc exec** pour exécuter une commande CLI Velero :

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> <command> <cr_name>
```

Exemple

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Option d'aide
L'option **velero --help** permet d'obtenir la liste de toutes les commandes CLI de Velero :

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  --help
```

Décrire la commande
Utilisez la commande **velero describe** pour obtenir un résumé des avertissements et des erreurs associés à un CR **Backup** ou **Restore**:

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> describe <cr_name>
```

Exemple

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Commande d'enregistrement
Utilisez la commande **velero logs** pour récupérer les journaux d'un CR **Backup** ou **Restore**:

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> logs <cr_name>
```

Exemple

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

4.5.5. Problèmes avec Velero et les webhooks d'admission

Velero a des capacités limitées pour résoudre les problèmes liés aux webhooks d'admission lors d'une restauration. Si vous avez des charges de travail avec des webhooks d'admission, vous devrez peut-être utiliser un plugin Velero supplémentaire ou modifier la façon dont vous restaurez la charge de travail.

Généralement, les charges de travail avec des webhooks d'admission exigent que vous créiez d'abord une ressource d'un type spécifique. Cela est particulièrement vrai si votre charge de travail comporte des ressources enfants, car les webhooks d'admission bloquent généralement les ressources enfants.

Par exemple, la création ou la restauration d'un objet de premier niveau tel que **service.serving.knative.dev** crée automatiquement des ressources enfants. Si vous procédez d'abord à

cette opération, vous ne devrez pas utiliser Velero pour créer et restaurer ces ressources. Cela permet d'éviter que les ressources enfants soient bloquées par un webhook d'admission que Velero pourrait utiliser.

4.5.5.1. Solutions de contournement pour la restauration des sauvegardes Velero qui utilisent des webhooks d'admission

Cette section décrit les étapes supplémentaires requises pour restaurer les ressources pour plusieurs types de sauvegardes Velero qui utilisent des webhooks d'admission.

4.5.5.1.1. Restaurer les ressources natives

Vous pouvez rencontrer des problèmes en utilisant Velero pour sauvegarder des ressources Knative qui utilisent des webhooks d'admission.

Vous pouvez éviter ces problèmes en restaurant d'abord la ressource **Service** de niveau supérieur chaque fois que vous sauvegardez et restaurez des ressources Knative qui utilisent des webhooks d'admission.

Procédure

- Restaurer la ressource de premier niveau **service.serving.knavtive.dev Service**:

```
$ velero restore <restore_name> \
  --from-backup=<backup_name> --include-resources \
  service.serving.knavtive.dev
```

4.5.5.1.2. Restauration des ressources IBM AppConnect

Si vous rencontrez des problèmes lorsque vous utilisez Velero pour restaurer une ressource IBM AppConnect dotée d'un webhook d'admission, vous pouvez effectuer les vérifications décrites dans cette procédure.

Procédure

1. Vérifiez si vous avez des plugins d'admission à la mutation de **kind: MutatingWebhookConfiguration** dans le cluster :

```
$ oc get mutatingwebhookconfigurations
```

2. Examinez le fichier YAML de chaque site **kind: MutatingWebhookConfiguration** pour vous assurer qu'aucune de ses règles ne bloque la création des objets qui posent problème. Pour plus d'informations, voir la documentation officielle de Kuberbetes .

3. Vérifiez que tout **spec.version** dans **type: Configuration.appconnect.ibm.com/v1beta1** utilisé au moment de la sauvegarde est pris en charge par l'opérateur installé.

Ressources complémentaires

- Plugins d'admission
- Plugins d'admission aux webhooks
- Types de plugins d'admission aux webhooks

4.5.6. Problèmes d'installation

Vous pouvez rencontrer des problèmes dus à l'utilisation de répertoires non valides ou d'informations d'identification incorrectes lors de l'installation de l'application de protection des données.

4.5.6.1. Le stockage de sauvegarde contient des répertoires non valides

Le journal du pod **Velero** affiche le message d'erreur **Backup storage contains invalid top-level directories**.

Cause

Le stockage d'objets contient des répertoires de premier niveau qui ne sont pas des répertoires Velero.

Solution

Si le stockage d'objets n'est pas dédié à Velero, vous devez spécifier un préfixe pour le seau en définissant le paramètre **spec.backupLocations.velero.objectStorage.prefix** dans le manifeste **DataProtectionApplication**.

4.5.6.2. Informations d'identification AWS incorrectes

Le journal du pod **oadp-aws-registry** affiche le message d'erreur, **InvalidAccessKeyId: The AWS Access Key Id you provided does not exist in our records.**

Le journal du pod **Velero** affiche le message d'erreur **NoCredentialProviders: no valid providers in chain**.

Cause

Le fichier **credentials-velero** utilisé pour créer l'objet **Secret** est mal formaté.

Solution

Assurez-vous que le fichier **credentials-velero** est correctement formaté, comme dans l'exemple suivant :

Exemple de fichier credentials-velero

```
[default] 1
aws_access_key_id=AKIAIOSFODNN7EXAMPLE 2
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

1 Profil par défaut AWS.

2 Ne mettez pas les valeurs entre guillemets (**''**, **'**).

4.5.7. Problèmes de sauvegarde et de restauration de la CR

Vous pouvez rencontrer ces problèmes courants avec les ressources personnalisées (CR) **Backup** et **Restore**.

4.5.7.1. Le CR de sauvegarde ne peut pas récupérer le volume

Le CR **Backup** affiche le message d'erreur **InvalidVolume.NotFound: The volume 'vol-xxxx' does not exist**.

Cause

Le volume persistant (PV) et les emplacements des instantanés se trouvent dans des régions différentes.

Solution

1. Modifiez la valeur de la clé **spec.snapshotLocations.velero.config.region** dans le manifeste **DataProtectionApplication** afin que l'emplacement de l'instantané se trouve dans la même région que le PV.
2. Créez un nouveau CR **Backup**.

4.5.7.2. L'état de la CR de sauvegarde reste en cours

Le statut d'une CR **Backup** reste dans la phase **InProgress** et ne s'achève pas.

Cause

Si une sauvegarde est interrompue, elle ne peut pas être reprise.

Solution

1. Récupérer les détails du CR **Backup**:

```
$ oc -n {namespace} exec deployment/velero -c velero -- ./velero \
  backup describe <backup>
```

2. Supprimer le CR **Backup**:

```
oc delete backup <backup> -n openshift-adp
```

   Il n'est pas nécessaire de nettoyer l'emplacement de sauvegarde car un CR **Backup** en cours n'a pas téléchargé de fichiers vers le stockage d'objets.

3. Créez un nouveau CR **Backup**.

4.5.7.3. Le statut de la CR de sauvegarde reste en PartiallyFailed (échec partiel)

L'état d'un CR **Backup** sans Restic en cours d'utilisation reste dans la phase **PartiallyFailed** et ne se termine pas. Un instantané du PVC affilié n'est pas créé.

Cause

Si la sauvegarde est créée sur la base de la classe d'instantané CSI, mais que l'étiquette est manquante, le plugin d'instantané CSI ne parvient pas à créer un instantané. En conséquence, le pod **Velero** enregistre une erreur similaire à la suivante :

```
time="2023-02-17T16:33:13Z" level=error msg="Error backing up item" backup=openshift-adp/user1-backup-check5 error="error executing custom action (groupResource=persistentvolumeclaims, namespace=busy1, name=pvc1-user1): rpc error: code = Unknown desc = failed to get volumesnapshotclass for storageclass ocs-storagecluster-ceph-rbd: failed to get
```

```
volumesnapshotclass for provisioner openshift-storage.rbd.csi.ceph.com, ensure that the desired volumesnapshot class has the velero.io/csi-volumesnapshot-class label" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=busybox-79799557b5-vprq
```

Solution

1. Supprimer le CR **Backup**:

```
oc delete backup <backup> -n openshift-adp
```

2. Si nécessaire, nettoyez les données stockées sur le site **BackupStorageLocation** pour libérer de l'espace.

3. Appliquez l'étiquette **velero.io/csi-volumesnapshot-class=true** à l'objet **VolumeSnapshotClass**:

```
oc label volumesnapshotclass/<snapclass_name> velero.io/csi-volumesnapshot-class=true
```

4. Créez un nouveau CR **Backup**.

4.5.8. Questions relatives à l'agriculture

Vous pouvez rencontrer ces problèmes lorsque vous sauvegardez des applications avec Restic.

4.5.8.1. Erreur d'autorisation de restic pour les volumes de données NFS avec l'option root_squash activée

Le journal du pod **Restic** affiche le message d'erreur : **controller=pod-volume-backup error="fork/exec/usr/bin/restic: permission denied"**.

Cause

Si vos volumes de données NFS ont activé **root_squash**, **Restic** est mappé à **nfsnobody** et n'a pas l'autorisation de créer des sauvegardes.

Solution

Vous pouvez résoudre ce problème en créant un groupe supplémentaire pour **Restic** et en ajoutant l'identifiant du groupe au manifeste **DataProtectionApplication**:

1. Créez un groupe supplémentaire pour **Restic** sur le volume de données NFS.

2. Activez le bit **setgid** sur les répertoires NFS pour que la propriété du groupe soit héritée.

3. Ajoutez le paramètre **spec.configuration.restic.supplementalGroups** et l'identifiant du groupe au manifeste **DataProtectionApplication**, comme dans l'exemple suivant :

```
spec:
  configuration:
    restic:
      enable: true
      supplementalGroups:
      - <group_id> 1
```

1 Indiquez l'ID du groupe supplémentaire.

4. Attendez que les pods **Restic** redémarrent pour que les changements soient appliqués.

4.5.8.2. Restic Backup CR ne peut pas être recréé après que le seau a été vidé

Si vous créez un CR Restic **Backup** pour un espace de noms, videz le seau de stockage d'objets, puis recréez le CR **Backup** pour le même espace de noms, le CR **Backup** recréé échoue.

Le journal du pod **velero** affiche le message d'erreur suivant : **stderr=Fatal: unable to open config file: Stat: The specified key does not exist.\nIs there a repository at the following location?**.

Cause

Velero ne recrée pas ou ne met pas à jour le référentiel Restic à partir du manifeste **ResticRepository** si les répertoires Restic sont supprimés du stockage d'objets. Voir le problème 4421 de Velero pour plus d'informations.

Solution

- Supprimez le référentiel Restic correspondant de l'espace de noms en exécutant la commande suivante :

```
oc delete resticrepository openshift-adp <name_of_the_restic_repository>
```

Dans le journal d'erreurs suivant, **mysql-persistent** est le dépôt Restic problématique. Le nom du dépôt apparaît en italique pour plus de clarté.

```
time="2021-12-29T18:29:14Z" level=info msg="1 errors
encountered backup up item" backup=velero/backup65
logSource="pkg/backup/backup.go:431" name=mysql-7d99fc949-qbkds
time="2021-12-29T18:29:14Z" level=error msg="Error backing up item"
backup=velero/backup65 error="pod volume backup failed: error running
restic backup, stderr=Fatal: unable to open config file: Stat: The
specified key does not exist.\nIs there a repository at the following
location?\ns3:http://minio-minio.apps.mayap-oadp-
veleo-1234.qe.devcluster.openshift.com/mayapvelerooadp2/velero1/
restic/mysql-persistent\n: exit status 1" error.file="/remote-source/
src/github.com/vmware-tanzu/velero/pkg/restic/backupper.go:184"
error.function="github.com/vmware-tanzu/velero/
pkg/restic.(*backupper).BackupPodVolumes"
logSource="pkg/backup/backup.go:435" name=mysql-7d99fc949-qbkds
```

4.5.9. Utilisation de l'outil de collecte obligatoire

Vous pouvez collecter des journaux, des mesures et des informations sur les ressources personnalisées de l'OADP à l'aide de l'outil **must-gather**.

Les données du site **must-gather** doivent être jointes à tous les dossiers clients.

Conditions préalables

- Vous devez être connecté au cluster OpenShift Container Platform en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous devez avoir installé OpenShift CLI (**oc**).

Procédure

1. Naviguez jusqu'au répertoire dans lequel vous souhaitez stocker les données **must-gather**.
2. Exécutez la commande **oc adm must-gather** pour l'une des options de collecte de données suivantes :

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1
```

Les données sont sauvegardées en tant que **must-gather/must-gather.tar.gz**. Vous pouvez télécharger ce fichier vers un dossier d'assistance sur le portail client de Red Hat .

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1 \
  -- /usr/bin/gather_metrics_dump
```

Cette opération peut prendre beaucoup de temps. Les données sont enregistrées sous **must-gather/metrics/prom_data.tar.gz**.

Visualisation des données de métrologie avec la console Prometheus
Vous pouvez consulter les données de mesure dans la console Prometheus.

Procédure

1. Décompressez le fichier **prom_data.tar.gz**:

```
$ tar -xvzf must-gather/metrics/prom_data.tar.gz
```

2. Créer une instance locale de Prometheus :

```
$ make prometheus-run
```

La commande affiche l'URL de Prometheus.

Sortie

```
Started Prometheus on http://localhost:9090
```

3. Lancez un navigateur web et accédez à l'URL pour visualiser les données à l'aide de la console web Prometheus.
4. Après avoir consulté les données, supprimez l'instance Prometheus et les données :

```
$ make prometheus-cleanup
```

4.6. API UTILISÉES AVEC L'OADP

Ce document fournit des informations sur les API suivantes que vous pouvez utiliser avec l'OADP :

- API Velero
- API OADP

4.6.1. API Velero

La documentation de l'API Velero est maintenue par Velero, et non par Red Hat. Elle peut être consultée à l'adresse Velero API types.

4.6.2. API OADP

Les tableaux suivants présentent la structure de l'API OADP :

Tableau 4.3. DataProtectionApplicationSpec

Propriété	Type	Description
backupLocations	[] **BackupLocation**	Définit la liste des configurations à utiliser pour **BackupStorageLocations**.
snapshotLocations	[] **SnapshotLocation**	Définit la liste des configurations à utiliser pour **VolumeSnapshotLocations**.
unsupportedOverrides	map [UnsupportedImageKey] string	Peut être utilisé pour remplacer les images dépendantes déployées pour le développement. Les options sont **veleroImageFqin**, **awsPluginImageFqin**, **openshiftPluginImageFqin**, **azurePluginImageFqin**, **gcpPluginImageFqin**, **csiPluginImageFqin**, **dataMoverImageFqin**, **resticRestoreImageFqin**, **kubevirtPluginImageFqin**, et **operator-type**.
podAnnotations	map [string] string	Utilisé pour ajouter des annotations aux pods déployés par les opérateurs.
podDnsPolicy	**DNSPolicy**	Définit la configuration du DNS d'un pod.
podDnsConfig	**PodDNSConfig**	Définit les paramètres DNS d'un pod en plus de ceux générés par **DNSPolicy**.
backupImages	*bool	Permet de spécifier si vous souhaitez ou non déployer un registre pour permettre la sauvegarde et la restauration des images.

Propriété	Type	Description
configuration	*ApplicationConfig	Utilisé pour définir la configuration du serveur de l'application de protection des données.
features	*Features	Définit la configuration du DPA pour activer les fonctions d'aperçu technologique.

Définitions complètes des schémas pour l'API de l'OADP .

Tableau 4.4. Lieu de sauvegarde

Propriété	Type	Description
velero	*velero.BackupStorageLocationSpec	Emplacement pour stocker les instantanés de volume, comme décrit dans Emplacement de stockage des sauvegardes.
bucket	*Emplacement du stockage en nuage	[Aperçu technologique] Automatise la création d'un bac chez certains fournisseurs de stockage dans le nuage pour l'utiliser comme emplacement de stockage de sauvegarde.

IMPORTANT

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

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

Définitions complètes du schéma pour le type **BackupLocation**.

Tableau 4.5. Localisation de l'instantané

Propriété	Type	Description

Propriété	Type	Description
velero	*VolumeSnapshotLocationSpec	Emplacement pour stocker les instantanés de volume, comme décrit dans Emplacement de l'instantané de volume.

Définitions complètes du schéma pour le type **SnapshotLocation**.

Tableau 4.6. ApplicationConfig

Propriété	Type	Description
velero	*VeleroConfig	Définit la configuration du serveur Velero.
restic	*ResticConfig	Définit la configuration du serveur Restic.

Définitions complètes du schéma pour le type **ApplicationConfig**.

Tableau 4.7. VeleroConfig

Propriété	Type	Description
featureFlags	[Chaîne de caractères	Définit la liste des fonctionnalités à activer pour l'instance Velero.
defaultPlugins	[Chaîne de caractères	Les types suivants de plugins Velero par défaut peuvent être installés : **aws**,**azure**, **csi**, **gcp**, **kubevirt**, et **openshift**.
customPlugins	[]CustomPlugin	Utilisé pour l'installation de plugins Velero personnalisés. Les plugins par défaut et personnalisés sont décrits dans les plugins OADP

Propriété	Type	Description
restoreResourcesVersionPriority	chaîne de caractères	Représente une carte de configuration qui est créée si elle est définie pour être utilisée avec l'indicateur de fonctionnalité **EnableAPIGroupVersions**. La définition de ce champ ajoute automatiquement **EnableAPIGroupVersions** à l'indicateur de fonctionnalité du serveur Velero.
noDefaultBackupLocation	bool	Pour installer Velero sans emplacement de stockage par défaut, vous devez définir l'option **noDefaultBackupLocation** afin de confirmer l'installation.
podConfig	*PodConfig	Définit la configuration du pod **Velero**.
logLevel	chaîne de caractères	Niveau de journalisation du serveur Velero (utilisez **debug** pour la journalisation la plus granulaire, laissez non défini pour la valeur par défaut de Velero). Les options valides sont **trace**, **debug**, **info**, **warning**, **error**, **fatal**, et **panic**.

Définitions complètes du schéma pour le type **VeleroConfig**.

Tableau 4.8. CustomPlugin

Propriété	Type	Description
name	chaîne de caractères	Nom du plugin personnalisé.
image	chaîne de caractères	Image du plugin personnalisé.

Définitions complètes du schéma pour le type **CustomPlugin**.

Tableau 4.9. ResticConfig

Propriété	Type	Description

Propriété	Type	Description
enable	*bool	Si la valeur est **true**, cela permet de sauvegarder et de restaurer les données à l'aide de Restic. Si la valeur est **false**, des instantanés sont nécessaires.
supplementalGroups	[]int64	Définit les groupes Linux à appliquer au pod **Restic**.
timeout	chaîne de caractères	Chaîne de durée fournie par l'utilisateur qui définit le délai d'attente de Restic. La valeur par défaut est **1hr** (1 heure). Une chaîne de durée est une séquence éventuellement signée de nombres décimaux, chacun avec une fraction facultative et un suffixe d'unité, comme **300ms**, -1.5h` ou **2h45m**. Les unités de temps valides sont **ns**, **us** (ou **µs**), **ms**, **s**, **m**, et **h**.
podConfig	***PodConfig**	Définit la configuration du pod **Restic**.

Définitions complètes du schéma pour le type **ResticConfig**.

Tableau 4.10. PodConfig

Propriété	Type	Description
nodeSelector	map [string] string	Définit le **nodeSelector** à fournir à un **Velero podSpec** ou un **Restic podSpec** .
tolerations	[]Tolérance	Définit la liste des tolérances à appliquer à un déploiement Velero ou à un Restic **daemonset**.
resourceAllocations	Besoins en ressources	Définissez les ressources spécifiques **limits** et **requests** pour un pod **Velero** ou un pod **Restic** comme décrit dans la section Définition des allocations de ressources CPU et mémoire de Velero.

Propriété	Type	Description
labels	map [string] string	Étiquettes à ajouter aux cosses.

Définitions complètes du schéma pour le type **PodConfig**.

Tableau 4.11. Caractéristiques

Propriété	Type	Description
dataMover	*DataMover	Définit la configuration du Data Mover.

Définitions complètes du schéma pour le type **Features**.

Tableau 4.12. DataMover

Propriété	Type	Description
enable	bool	S'il est défini sur **true**, il déploie le contrôleur de transfert d'instantanés de volume et un plugin CSI Data Mover modifié. Si la valeur est **false**, ces éléments ne sont pas déployés.
credentialName	chaîne de caractères	Nom Restic **Secret** fourni par l'utilisateur pour Data Mover.
timeout	chaîne de caractères	Chaîne de durée fournie par l'utilisateur pour l'achèvement de **VolumeSnapshotBackup** et **VolumeSnapshotRestore**. La valeur par défaut est **10m** (10 minutes). Une chaîne de durée est une séquence éventuellement signée de nombres décimaux, chacun avec une fraction optionnelle et un suffixe d'unité, comme **300ms**, -1.5h` ou **2h45m**. Les unités de temps valides sont **ns**, **us** (ou **µs**), **ms**, **s**, **m**, et **h**.

L'API de l'OADP est décrite plus en détail dans OADP Operator.

4.7. CARACTÉRISTIQUES ET FONCTIONNALITÉS AVANCÉES DE L'OADP

Ce document fournit des informations sur les caractéristiques et les fonctionnalités avancées d'OpenShift API for Data Protection (OADP).

4.7.1. Travailler avec différentes versions de l'API Kubernetes sur le même cluster

4.7.1.1. Liste des versions des groupes de l'API Kubernetes sur un cluster

Un groupe de sources peut proposer plusieurs versions d'une API, l'une de ces versions étant la version préférée de l'API. Par exemple, un groupe de sources avec une API nommée **Example** peut être disponible dans les groupes d'API **example.com/v1** et **example.com/v1beta2**.

Si vous utilisez Velero pour sauvegarder et restaurer un tel cluster source, Velero ne sauvegarde que la version de cette ressource qui utilise la version préférée de son API Kubernetes.

Pour revenir à l'exemple ci-dessus, si **example.com/v1** est l'API préférée, Velero ne sauvegarde que la version d'une ressource qui utilise **example.com/v1**. En outre, le cluster cible doit avoir enregistré **example.com/v1** dans son ensemble de ressources API disponibles pour que Velero puisse restaurer la ressource sur le cluster cible.

Par conséquent, vous devez générer une liste des versions du groupe d'API Kubernetes sur votre cluster cible pour vous assurer que la version d'API préférée est enregistrée dans son ensemble de ressources d'API disponibles.

Procédure

- Entrez la commande suivante :

```
$ oc api-resources
```

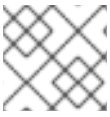
4.7.1.2. À propos de l'activation des versions des groupes d'API

Par défaut, Velero ne sauvegarde que les ressources qui utilisent la version préférée de l'API Kubernetes. Cependant, Velero inclut également une fonctionnalité, Enable API Group Versions , qui permet de surmonter cette limitation. Lorsqu'elle est activée sur le cluster source, cette fonctionnalité permet à Velero de sauvegarder *all* les versions du groupe API Kubernetes qui sont prises en charge sur le cluster, et pas seulement la version préférée. Une fois les versions stockées dans le fichier .tar de sauvegarde, elles sont disponibles pour être restaurées sur le cluster de destination.

Par exemple, un cluster source avec une API nommée **Example** peut être disponible dans les groupes d'API **example.com/v1** et **example.com/v1beta2**, **example.com/v1** étant l'API préférée.

Si la fonction Activer les versions du groupe API n'est pas activée, Velero ne sauvegarde que la version préférée du groupe API pour **Example**, c'est-à-dire **example.com/v1**. Si cette fonctionnalité est activée, Velero sauvegarde également **example.com/v1beta2**.

Lorsque la fonctionnalité Enable API Group Versions est activée sur le cluster de destination, Velero sélectionne la version à restaurer en fonction de l'ordre de priorité des versions des groupes d'API.



NOTE

Enable API Group Versions est encore en version bêta.

Velero utilise l'algorithme suivant pour attribuer des priorités aux versions de l'API, **1** étant la priorité absolue :

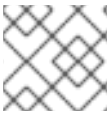
1. Version préférée du cluster *destination*
2. Version préférée du cluster source
3. Version commune non préférée prise en charge avec la priorité de version Kubernetes la plus élevée

Ressources complémentaires

- Activer la fonction Versions des groupes d'API

4.7.1.3. Utilisation de l'activation des versions des groupes API

Vous pouvez utiliser la fonctionnalité Enable API Group Versions de Velero pour sauvegarder *all* Kubernetes API group versions that are supported on a cluster, not only the preferred one.



NOTE

Enable API Group Versions est encore en version bêta.

Procédure

- Configurez l'indicateur de fonctionnalité **EnableAPIGroupVersions**:

```
apiVersion: oadp.openshift.io/vialpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      featureFlags:
      - EnableAPIGroupVersions
```

Ressources complémentaires

- Activer la fonction Versions des groupes d'API

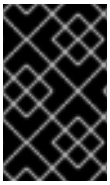
CHAPITRE 5. SAUVEGARDE ET RESTAURATION DU PLAN DE CONTRÔLE

5.1. SAUVEGARDE DE ETCD

etcd est le magasin clé-valeur d'OpenShift Container Platform, qui conserve l'état de tous les objets de ressources.

Sauvegardez régulièrement les données etcd de votre cluster et stockez-les dans un endroit sûr, idéalement en dehors de l'environnement OpenShift Container Platform. Ne prenez pas de sauvegarde etcd avant la fin de la première rotation des certificats, qui a lieu 24 heures après l'installation, sinon la sauvegarde contiendra des certificats expirés. Il est également recommandé d'effectuer des sauvegardes etcd en dehors des heures de pointe, car l'instantané etcd a un coût d'E/S élevé.

Veillez à effectuer une sauvegarde etcd après avoir mis à niveau votre cluster. Ceci est important car lorsque vous restaurez votre cluster, vous devez utiliser une sauvegarde etcd qui a été prise à partir de la même version de z-stream. Par exemple, un cluster OpenShift Container Platform 4.y.z doit utiliser une sauvegarde etcd provenant de la version 4.y.z.



IMPORTANT

Sauvegardez les données etcd de votre cluster en effectuant une seule invocation du script de sauvegarde sur un hôte du plan de contrôle. Ne faites pas de sauvegarde pour chaque hôte du plan de contrôle.

Après avoir effectué une sauvegarde d'etcd, vous pouvez restaurer un état antérieur du cluster.

5.1.1. Sauvegarde des données etcd

Suivez ces étapes pour sauvegarder les données etcd en créant un instantané etcd et en sauvegardant les ressources des pods statiques. Cette sauvegarde peut être enregistrée et utilisée ultérieurement si vous avez besoin de restaurer etcd.

IMPORTANT

N'effectuez une sauvegarde qu'à partir d'un seul hôte de plan de contrôle. N'effectuez pas de sauvegarde à partir de chaque hôte de plan de contrôle du cluster.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez vérifié si le proxy à l'échelle du cluster est activé.

ASTUCE

Vous pouvez vérifier si le proxy est activé en examinant la sortie de **oc get proxy cluster -o yaml**. Le proxy est activé si les champs **httpProxy**, **httpsProxy** et **noProxy** ont des valeurs définies.

Procédure

1. Démarrer une session de débogage pour un nœud de plan de contrôle :

```
oc debug node/<node_name>
```

2. Changez votre répertoire racine en /**host**:

```
sh-4.2# chroot /host
```

3. Si le proxy à l'échelle du cluster est activé, assurez-vous que vous avez exporté les variables d'environnement **NO_PROXY**, **HTTP_PROXY**, et **HTTPS_PROXY**.

4. Exécutez le script **cluster-backup.sh** et indiquez l'emplacement où sauvegarder la sauvegarde.

ASTUCE

Le script **cluster-backup.sh** est maintenu en tant que composant de l'opérateur de cluster etcd et est une enveloppe autour de la commande **etcdctl snapshot save**.

```
sh-4.4# /usr/local/bin/cluster-backup.sh /home/core/assets/backup
```

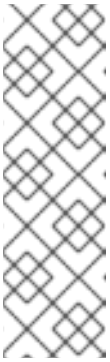
Exemple de sortie de script

```
found latest kube-apiserver: /etc/kubernetes/static-pod-resources/kube-apiserver-pod-6
found latest kube-controller-manager: /etc/kubernetes/static-pod-resources/kube-controller-manager-pod-7
found latest kube-scheduler: /etc/kubernetes/static-pod-resources/kube-scheduler-pod-6
found latest etcd: /etc/kubernetes/static-pod-resources/etcd-pod-3
ede95fe6b88b87ba86a03c15e669fb4aa5bf0991c180d3c6895ce72eaade54a1
etcdctl version: 3.4.14
API version: 3.4
{"level":"info","ts":1624647639.0188997,"caller":"snapshot/v3_snapshot.go:119","msg":"created temporary db file","path":"/home/core/assets/backup/snapshot_2021-06-25_190035.db.part"}
{"level":"info","ts":"2021-06-25T19:00:39.030Z","caller":"clientv3/maintenance.go:200","msg":"opened snapshot stream; downloading"}
{"level":"info","ts":1624647639.0301006,"caller":"snapshot/v3_snapshot.go:127","msg":"fetching snapshot","endpoint":"https://10.0.0.5:2379"}
{"level":"info","ts":"2021-06-25T19:00:40.215Z","caller":"clientv3/maintenance.go:208","msg":"completed snapshot read; closing"}
{"level":"info","ts":1624647640.6032252,"caller":"snapshot/v3_snapshot.go:142","msg":"fetched snapshot","endpoint":"https://10.0.0.5:2379","size":"114 MB","took":1.584090459}
{"level":"info","ts":1624647640.6047094,"caller":"snapshot/v3_snapshot.go:152","msg":"saved","path":"/home/core/assets/backup/snapshot_2021-06-25_190035.db"}
Snapshot saved at /home/core/assets/backup/snapshot_2021-06-25_190035.db
{"hash":3866667823,"revision":31407,"totalKey":12828,"totalSize":114446336}
snapshot db and kube resources are successfully saved to /home/core/assets/backup
```

Dans cet exemple, deux fichiers sont créés dans le répertoire /**home**/**core**/**assets**/**backup**/ sur l'hôte du plan de contrôle :

- **snapshot_<datetimestamp>.db**: Ce fichier est le snapshot etcd. Le script **cluster-backup.sh** confirme sa validité.

- **static_kuberesources_<datetimestamp>.tar.gz**: Ce fichier contient les ressources pour les pods statiques. Si le chiffrement etcd est activé, il contient également les clés de chiffrement pour l'instantané etcd.



NOTE

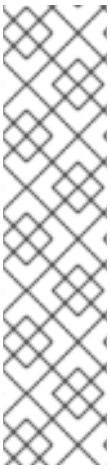
Si le cryptage etcd est activé, il est recommandé de stocker ce deuxième fichier séparément de l'instantané etcd pour des raisons de sécurité. Toutefois, ce fichier est nécessaire pour restaurer à partir de l'instantané etcd.

Gardez à l'esprit que le chiffrement etcd ne chiffre que les valeurs, pas les clés. Cela signifie que les types de ressources, les espaces de noms et les noms d'objets ne sont pas chiffrés.

5.2. REMPLACEMENT D'UN MEMBRE ETCD MALSAIN

Ce document décrit le processus de remplacement d'un membre etcd malsain.

Ce processus dépend du fait que le membre etcd est malsain parce que la machine ne fonctionne pas ou que le nœud n'est pas prêt, ou qu'il est malsain parce que le pod etcd est en train de faire du crashlooping.



NOTE

Si vous avez perdu la majorité de vos hôtes du plan de contrôle, suivez la procédure de reprise après sinistre pour restaurer un état antérieur du cluster au lieu de suivre cette procédure.

Si les certificats du plan de contrôle ne sont pas valides sur le membre remplacé, vous devez suivre la procédure de récupération des certificats de plan de contrôle expirés au lieu de cette procédure.

Si un nœud de plan de contrôle est perdu et qu'un nouveau est créé, l'opérateur de cluster etcd se charge de générer les nouveaux certificats TLS et d'ajouter le nœud en tant que membre etcd.

5.2.1. Conditions préalables

- Effectuez une sauvegarde d'etcd avant de remplacer un membre d'etcd malsain.

5.2.2. Identification d'un membre etcd malsain

Vous pouvez identifier si votre cluster a un membre etcd en mauvaise santé.

Conditions préalables

- Accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

1. Vérifiez l'état de la condition d'état **EtcdMembersAvailable** à l'aide de la commande suivante :

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
```

2. Examiner les résultats :

```
2 of 3 members are available, ip-10-0-131-183.ec2.internal is unhealthy
```

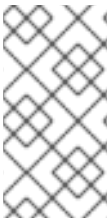
Cet exemple montre que le membre **ip-10-0-131-183.ec2.internal** etcd n'est pas sain.

5.2.3. Déterminer l'état d'un membre etcd malsain

Les étapes pour remplacer un membre etcd malsain dépendent de l'état dans lequel se trouve votre membre etcd :

- La machine ne fonctionne pas ou le nœud n'est pas prêt
- Le pod etcd est en train de faire un crashloop

Cette procédure permet de déterminer dans quel état se trouve votre membre etcd. Cela vous permet de savoir quelle procédure suivre pour remplacer le membre etcd en mauvais état.



NOTE

Si vous savez que la machine ne fonctionne pas ou que le nœud n'est pas prêt, mais que vous vous attendez à ce qu'il revienne bientôt à un état sain, vous n'avez pas besoin d'exécuter une procédure pour remplacer le membre etcd. L'opérateur de cluster etcd se synchronisera automatiquement lorsque la machine ou le nœud redeviendra sain.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez identifié un membre etcd malsain.

Procédure

1. Déterminer si le site **machine is not running**

```
$ oc get machines -A -ojsonpath='{range .items[*]}{@.status.nodeRef.name}
{@.status.providerStatus.instanceState}{@.status.providerStatus.instanceState}{\N-"\N-"\N"}"
| grep -v running
```

Exemple de sortie

```
ip-10-0-131-183.ec2.internal  stopped (1)
```

(1) Cette sortie indique le nœud et l'état de la machine du nœud. Si l'état est différent de **running**, alors la page **machine is not running**

Si le site **machine is not running**, suivre la procédure *Replacing an unhealthy etcd member whose machine is not running or whose node is not ready*.

2. Déterminer si le site **node is not ready**.

Si l'un ou l'autre des scénarios suivants est vrai, alors le site **node is not ready**.

- Si la machine fonctionne, vérifiez si le nœud est inaccessible :

```
$ oc get nodes -o jsonpath='{range .items[*]}{"\n"}{.metadata.name}{"\t"}{range
.spec.taints[*]}{.key}{" "}' | grep unreachable
```

Exemple de sortie

```
ip-10-0-131-183.ec2.internal node-role.kubernetes.io/master
node.kubernetes.io/unreachable node.kubernetes.io/unreachable 1
```

1 Si le nœud est répertorié avec une tare **unreachable**, alors la tare **node is not ready**.

- Si le nœud est toujours accessible, vérifiez si le nœud est répertorié comme **NotReady**:

```
$ oc get nodes -l node-role.kubernetes.io/master | grep "NotReady"
```

Exemple de sortie

```
ip-10-0-131-183.ec2.internal   NotReady   master   122m   v1.25.0 1
```

1 Si le nœud est répertorié comme **NotReady**, alors le nœud **node is not ready**.

Si le site **node is not ready**, suivre la procédure *Replacing an unhealthy etcd member whose machine is not running or whose node is not ready*.

3. Déterminer si le site **etcd pod is crashlooping**
   Si la machine fonctionne et que le nœud est prêt, vérifiez si le pod etcd fait du crashlooping.

   a. Vérifiez que tous les nœuds du plan de contrôle sont répertoriés comme **Ready**:

```
$ oc get nodes -l node-role.kubernetes.io/master
```

Exemple de sortie

```
NAME                           STATUS   ROLES    AGE     VERSION
ip-10-0-131-183.ec2.internal   Ready    master   6h13m   v1.25.0
ip-10-0-164-97.ec2.internal    Ready    master   6h13m   v1.25.0
ip-10-0-154-204.ec2.internal   Ready    master   6h13m   v1.25.0
```

   b. Vérifier si l'état d'un pod etcd est **Error** ou **CrashloopBackoff**:

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

Exemple de sortie

```
etcd-ip-10-0-131-183.ec2.internal          2/3     Error     7          6h9m 1
etcd-ip-10-0-164-97.ec2.internal           3/3     Running   0          6h6m
etcd-ip-10-0-154-204.ec2.internal          3/3     Running   0          6h6m
```

1 Puisque le statut de ce pod est **Error**, alors le **etcd pod is crashlooping**

Si le site **etcd pod is crashlooping** suivre la procédure *Replacing an unhealthy etcd member whose etcd pod is crashlooping*.

5.2.4. Remplacement d'un membre etcd malsain

Selon l'état de votre membre etcd malsain, utilisez l'une des procédures suivantes :

- Remplacement d'un membre etcd en mauvaise santé dont la machine ne fonctionne pas ou dont le nœud n'est pas prêt
- Remplacement d'un membre etcd en mauvaise santé dont le pod etcd est en crashlooping
- Remplacement d'un membre etcd baremetal arrêté et malsain

5.2.4.1. Remplacement d'un membre etcd en mauvaise santé dont la machine ne fonctionne pas ou dont le nœud n'est pas prêt

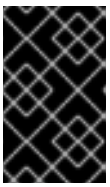
Cette procédure détaille les étapes à suivre pour remplacer un membre etcd qui n'est pas sain, soit parce que la machine ne fonctionne pas, soit parce que le nœud n'est pas prêt.

NOTE

Si votre cluster utilise un jeu de machines de plan de contrôle, voir "Récupération d'un opérateur etcd dégradé" dans "Dépannage du jeu de machines de plan de contrôle" pour une procédure de récupération etcd plus simple.

Conditions préalables

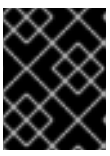
- Vous avez identifié le membre etcd malsain.
- Vous avez vérifié que la machine n'est pas en cours d'exécution ou que le nœud n'est pas prêt.



IMPORTANT

Vous devez attendre si les autres nœuds du plan de contrôle sont hors tension. Les nœuds du plan de contrôle doivent rester hors tension jusqu'à ce que le remplacement d'un membre etcd malsain soit terminé.

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez effectué une sauvegarde etcd.



IMPORTANT

Il est important de faire une sauvegarde d'etcd avant d'effectuer cette procédure afin de pouvoir restaurer votre cluster en cas de problème.

Procédure

1. Retirer l'élément malsain.

a. Choisissez un pod qui est *not* sur le nœud affecté :
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

Exemple de sortie

```
etcd-ip-10-0-131-183.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-164-97.ec2.internal              3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal             3/3     Running     0          124m
```

b. Se connecter au conteneur etcd en cours d'exécution, en passant le nom d'un pod qui n'est pas sur le nœud affecté :
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

c. Consulter la liste des membres :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+------------------+---------+-----------------------------+---------------------------+-----------------
-----------+
|        ID        | STATUS  |             NAME            |        PEER ADDRS         |       CLIENT
ADDRS        |
+------------------+---------+-----------------------------+---------------------------+-----------------
-----------+
| 6fc1e7c9db35841d | started | ip-10-0-131-183.ec2.internal | https://10.0.131.183:2380 |
https://10.0.131.183:2379 |
| 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
https://10.0.154.204:2379 |
+------------------+---------+-----------------------------+---------------------------+-----------------
-----------+
```

Notez l'ID et le nom du membre etcd malsain, car ces valeurs seront nécessaires plus tard dans la procédure. La commande **$ etcdctl endpoint health** listera le membre supprimé jusqu'à ce que la procédure de remplacement soit terminée et qu'un nouveau membre soit ajouté.

d. Supprimez le membre etcd malsain en fournissant l'ID à la commande **etcdctl member remove**:

```
sh-4.2# etcdctl member remove 6fc1e7c9db35841d
```

Exemple de sortie

```
Member 6fc1e7c9db35841d removed from cluster ead669ce1fbfb346
```

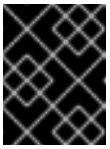
e. Consultez à nouveau la liste des membres et vérifiez que le membre a bien été supprimé :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+------------------+---------+----------------------------+--------------------------+----------------
----------+
|        ID        | STATUS  |            NAME            |       PEER ADDRS         |      CLIENT
ADDRS       |
+------------------+---------+----------------------------+--------------------------+----------------
----------+
| 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
https://10.0.154.204:2379 |
+------------------+---------+----------------------------+--------------------------+----------------
----------+
```

Vous pouvez maintenant quitter le shell du nœud.



IMPORTANT

Après avoir supprimé le membre, le cluster peut être inaccessible pendant une courte période, le temps que les instances etcd restantes redémarrent.

2. Désactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

Cette commande permet de s'assurer que vous pouvez recréer les secrets et déployer les pods statiques avec succès.

3. Supprime les anciens secrets du membre etcd malsain qui a été supprimé.

a. Liste les secrets du membre etcd malsain qui a été supprimé.

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

1 Saisissez le nom du membre etcd malsain dont vous avez pris note plus tôt dans cette procédure.

Il y a un pair, un serveur et un secret de métrique, comme le montre la sortie suivante :

Exemple de sortie

```
etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                 2      47m
etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                 2      47m
etcd-serving-metrics-ip-10-0-131-183.ec2.internal  kubernetes.io/tls                 2
```

```
47m
```

b. Supprime les secrets du membre etcd malsain qui a été supprimé.

i. Supprimer le secret de l'homologue :

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

ii. Supprimer le secret de service :

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

iii. Supprimer le secret des métriques :

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-183.ec2.internal
```

4. Supprimer et recréer la machine du plan de contrôle. Une fois cette machine recréée, une nouvelle révision est forcée et etcd se met automatiquement à l'échelle.
Si vous utilisez une infrastructure fournie par l'installateur ou si vous avez utilisé l'API Machine pour créer vos machines, suivez ces étapes. Sinon, vous devez créer le nouveau master en utilisant la même méthode que celle utilisée pour le créer à l'origine.

a. Obtenir la machine pour le membre en mauvaise santé.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-0          Running   m4.xlarge   us-east-1   us-east-1a
3h37m   ip-10-0-131-183.ec2.internal   aws:///us-east-1a/i-0ec2782f8287dfb7e   stopped
1
clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-
1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b
3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-
1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a
running
```

1 Il s'agit de la machine du plan de contrôle pour le nœud malsain, **ip-10-0-131-183.ec2.internal**.

b. Enregistrez la configuration de la machine dans un fichier sur votre système de fichiers :

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 Indiquez le nom de la machine du plan de contrôle pour le nœud malsain.

c. Modifiez le fichier **new-master-machine.yaml** créé à l'étape précédente pour lui attribuer un nouveau nom et supprimer les champs inutiles.

i. Retirer toute la section **status**:

```
status:
  addresses:
  - address: 10.0.131.183
    type: InternalIP
  - address: ip-10-0-131-183.ec2.internal
    type: InternalDNS
  - address: ip-10-0-131-183.ec2.internal
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Node
    name: ip-10-0-131-183.ec2.internal
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: awsproviderconfig.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: AWSMachineProviderStatus
```

ii. Changez le nom du champ **metadata.name**.
Il est recommandé de conserver le même nom de base que l'ancienne machine et de remplacer le numéro de fin par le prochain numéro disponible. Dans cet exemple, **clustername-8qw5l-master-0** est remplacé par **clustername-8qw5l-master-3**.

Par exemple :

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. Supprimer le champ **spec.providerID**:

```
providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

d. Supprimez l'objet **BareMetalHost** en exécutant la commande suivante, en remplaçant **<host_name>** par le nom de l'hôte à nu du nœud malsain :

```
oc delete bmh -n openshift-machine-api <host_name>
```

e. Supprimez la machine du membre malsain en exécutant la commande suivante, en remplaçant **<machine_name>** par le nom de la machine du plan de contrôle du nœud malsain, par exemple **clustername-8qw5l-master-0**:

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

f. Vérifiez que la machine a été supprimée :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                                 PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1           Running   m4.xlarge   us-east-1   us-east-1b   3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2           Running   m4.xlarge   us-east-1   us-east-1c   3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a   running
```

g. Créez la nouvelle machine à l'aide du fichier **new-master-machine.yaml**:

```
$ oc apply -f new-master-machine.yaml
```

h. Vérifiez que la nouvelle machine a été créée :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                                 PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1           Running   m4.xlarge   us-east-1   us-east-1b   3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2           Running   m4.xlarge   us-east-1   us-east-1c   3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-master-3           Provisioning   m4.xlarge   us-east-1   us-east-1a   85s   ip-10-0-133-53.ec2.internal   aws:///us-east-1a/i-015b0888fe17bc2c8   running
```

```
1
clustername-8qw5l-worker-us-east-1a-wbtgd  Running      m4.large  us-east-1  us-east-1a  3h28m  ip-10-0-129-226.ec2.internal  aws:///us-east-1a/i-010ef6279b4662ced  running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running      m4.large  us-east-1  us-east-1b  3h28m  ip-10-0-144-248.ec2.internal  aws:///us-east-1b/i-0cb45ac45a166173b  running
clustername-8qw5l-worker-us-east-1c-pkg26  Running      m4.large  us-east-1  us-east-1c  3h28m  ip-10-0-170-181.ec2.internal  aws:///us-east-1c/i-06861c00007751b0a  running
```

1 La nouvelle machine, **clustername-8qw5l-master-3**, est en cours de création et sera prête lorsque la phase passera de **Provisioning** à **Running**.

La création de la nouvelle machine peut prendre quelques minutes. L'opérateur du cluster etcd se synchronisera automatiquement lorsque la machine ou le nœud reviendra à un état sain.

5. Réactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

6. Vous pouvez vérifier que la section **unsupportedConfigOverrides** est supprimée de l'objet en entrant cette commande :

```
$ oc get etcd/cluster -oyaml
```

7. Si vous utilisez OpenShift à nœud unique, redémarrez le nœud. Sinon, vous risquez de rencontrer l'erreur suivante dans l'opérateur de cluster etcd :

Exemple de sortie

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

Vérification

1. Vérifiez que tous les pods etcd fonctionnent correctement.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

Exemple de sortie

```
etcd-ip-10-0-133-53.ec2.internal             3/3     Running     0          7m49s
etcd-ip-10-0-164-97.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal            3/3     Running     0          124m
```

Si la sortie de la commande précédente n'indique que deux pods, vous pouvez forcer manuellement un redéploiement d'etcd. Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc patch etcd cluster -p='{"spec" : {\i1}"forceRedeploymentReason" : \N-"recovery-'\N"$(
date --rfc-3339=ns )'\N'\N"}" -type=merge} --type=merge 1
```

1 La valeur **forceRedeploymentReason** doit être unique, c'est pourquoi un horodatage est ajouté.

2. Vérifiez qu'il y a exactement trois membres etcd.

   a. Se connecter au conteneur etcd en cours d'exécution, en passant le nom d'un pod qui n'était pas sur le nœud affecté :
   Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

   ```
   $ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
   ```

   b. Consulter la liste des membres :

   ```
   sh-4.2# etcdctl member list -w table
   ```

   Exemple de sortie

   ```
   +------------------+---------+-----------------------------+---------------------------+---------------
   -----------+
   |        ID        | STATUS  |             NAME            |        PEER ADDRS         |      CLIENT
   ADDRS       |
   +------------------+---------+-----------------------------+---------------------------+---------------
   -----------+
   | 5eb0d6b8ca24730c | started |  ip-10-0-133-53.ec2.internal |  https://10.0.133.53:2380 |
   https://10.0.133.53:2379 |
   | 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
   https://10.0.164.97:2379 |
   | ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
   https://10.0.154.204:2379 |
   +------------------+---------+-----------------------------+---------------------------+---------------
   -----------+
   ```

   Si la sortie de la commande précédente répertorie plus de trois membres etcd, vous devez supprimer avec précaution le membre indésirable.

> **AVERTISSEMENT**
>
> Veillez à supprimer le bon membre etcd ; la suppression d'un bon membre etcd peut entraîner une perte de quorum.

Ressources complémentaires

- Récupération d'un opérateur etcd dégradé

5.2.4.2. Remplacement d'un membre etcd en mauvaise santé dont le pod etcd est en crashlooping

Cette procédure détaille les étapes à suivre pour remplacer un membre etcd qui n'est pas en bonne santé parce que le pod etcd fait du crashlooping.

Conditions préalables

- Vous avez identifié le membre etcd malsain.
- Vous avez vérifié que le pod etcd fait du crashlooping.
- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez effectué une sauvegarde etcd.

IMPORTANT

Il est important de faire une sauvegarde d'etcd avant d'effectuer cette procédure afin de pouvoir restaurer votre cluster en cas de problème.

Procédure

1. Arrêter le pod etcd de crashlooping.
   a. Déboguer le nœud qui fait du crashlooping.
   Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
oc debug node/ip-10-0-131-183.ec2.internal 1
```

   1 Remplacez ce nom par le nom du nœud malsain.

   b. Changez votre répertoire racine en /**host**:

```
sh-4.2# chroot /host
```

   c. Déplacer le fichier pod etcd existant hors du répertoire kubelet manifest :

```
sh-4.2# mkdir /var/lib/etcd-backup
```

```
sh-4.2# mv /etc/kubernetes/manifests/etcd-pod.yaml /var/lib/etcd-backup/
```

   d. Déplacez le répertoire de données etcd vers un autre emplacement :

```
sh-4.2# mv /var/lib/etcd/ /tmp
```

   Vous pouvez maintenant quitter le shell du nœud.

2. Retirer l'élément malsain.

   a. Choisissez un pod qui est *not* sur le nœud affecté.
   Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

   ```
   $ oc -n openshift-etcd get pods -l k8s-app=etcd
   ```

   Exemple de sortie

   ```
   etcd-ip-10-0-131-183.ec2.internal             2/3     Error       7          6h9m
   etcd-ip-10-0-164-97.ec2.internal              3/3     Running     0          6h6m
   etcd-ip-10-0-154-204.ec2.internal             3/3     Running     0          6h6m
   ```

   b. Se connecter au conteneur etcd en cours d'exécution, en passant le nom d'un pod qui n'est pas sur le nœud affecté.
   Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

   ```
   $ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
   ```

   c. Consulter la liste des membres :

   ```
   sh-4.2# etcdctl member list -w table
   ```

   Exemple de sortie

   ```
   +------------------+---------+------------------------------+---------------------------+---------------
   -----------+
   |        ID        | STATUS  |             NAME             |        PEER ADDRS         |      CLIENT
   ADDRS       |
   +------------------+---------+------------------------------+---------------------------+---------------
   -----------+
   | 62bcf33650a7170a | started | ip-10-0-131-183.ec2.internal | https://10.0.131.183:2380 |
   https://10.0.131.183:2379 |
   | b78e2856655bc2eb | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
   https://10.0.164.97:2379 |
   | d022e10b498760d5 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380
   | https://10.0.154.204:2379 |
   +------------------+---------+------------------------------+---------------------------+---------------
   -----------+
   ```

   Notez l'ID et le nom du membre etcd malsain, car ces valeurs seront nécessaires plus tard dans la procédure.

   d. Supprimez le membre etcd malsain en fournissant l'ID à la commande **etcdctl member remove**:

   ```
   sh-4.2# etcdctl member remove 62bcf33650a7170a
   ```

   Exemple de sortie

   ```
   Member 62bcf33650a7170a removed from cluster ead669ce1fbfb346
   ```

e. Consultez à nouveau la liste des membres et vérifiez que le membre a bien été supprimé :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+------------------+---------+----------------------------+--------------------------+----------------
----------+
|        ID        | STATUS  |            NAME            |       PEER ADDRS         |      CLIENT
ADDRS       |
+------------------+---------+----------------------------+--------------------------+----------------
----------+
| b78e2856655bc2eb | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| d022e10b498760d5 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380
| https://10.0.154.204:2379 |
+------------------+---------+----------------------------+--------------------------+----------------
----------+
```

Vous pouvez maintenant quitter le shell du nœud.

3. Désactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

Cette commande permet de s'assurer que vous pouvez recréer les secrets et déployer les pods statiques avec succès.

4. Supprime les anciens secrets du membre etcd malsain qui a été supprimé.

a. Liste les secrets du membre etcd malsain qui a été supprimé.

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

1 Saisissez le nom du membre etcd malsain dont vous avez pris note plus tôt dans cette procédure.

Il y a un pair, un serveur et un secret de métrique, comme le montre la sortie suivante :

Exemple de sortie

```
etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                    2      47m
etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                    2      47m
etcd-serving-metrics-ip-10-0-131-183.ec2.internal   kubernetes.io/tls                    2
47m
```

b. Supprime les secrets du membre etcd malsain qui a été supprimé.

i. Supprimer le secret de l'homologue :

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

ii. Supprimer le secret de service :

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

iii. Supprimer le secret des métriques :

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-183.ec2.internal
```

5. Forcer le redéploiement de etcd.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc patch etcd cluster -p='{"spec" : {\i1}"forceRedeploymentReason" : \N "single-master-recovery-'\N"$( date --rfc-3339=ns )'\N"}" -type=merge} --type=merge 1
```

1 La valeur **forceRedeploymentReason** doit être unique, c'est pourquoi un horodatage est ajouté.

Lorsque l'opérateur de cluster etcd effectue un redéploiement, il s'assure que tous les nœuds du plan de contrôle disposent d'un pod etcd fonctionnel.

6. Réactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

7. Vous pouvez vérifier que la section **unsupportedConfigOverrides** est supprimée de l'objet en entrant cette commande :

```
$ oc get etcd/cluster -oyaml
```

8. Si vous utilisez OpenShift à nœud unique, redémarrez le nœud. Sinon, vous risquez de rencontrer l'erreur suivante dans l'opérateur de cluster etcd :

Exemple de sortie

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

Vérification

- Vérifier que le nouveau membre est disponible et en bonne santé.

a. Connectez-vous à nouveau au conteneur etcd en cours d'exécution.
Dans un terminal ayant accès au cluster en tant qu'utilisateur cluster-admin, exécutez la commande suivante :

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

b. Vérifier que tous les membres sont en bonne santé :

```
sh-4.2# etcdctl endpoint health
```

Exemple de sortie

```
https://10.0.131.183:2379 is healthy: successfully committed proposal: took =
16.671434ms
https://10.0.154.204:2379 is healthy: successfully committed proposal: took =
16.698331ms
https://10.0.164.97:2379 is healthy: successfully committed proposal: took =
16.621645ms
```

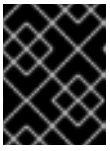
5.2.4.3. Remplacement d'un membre etcd bare metal en mauvaise santé dont la machine ne fonctionne pas ou dont le nœud n'est pas prêt

Cette procédure détaille les étapes à suivre pour remplacer un membre etcd en métal nu qui n'est pas en bonne santé, soit parce que la machine ne fonctionne pas, soit parce que le nœud n'est pas prêt.

Si vous exécutez une infrastructure fournie par l'installateur ou si vous avez utilisé l'API Machine pour créer vos machines, suivez ces étapes. Sinon, vous devez créer le nouveau nœud de plan de contrôle en utilisant la même méthode que celle utilisée pour le créer à l'origine.

Conditions préalables

- Vous avez identifié le membre malsain de bare metal etcd.
- Vous avez vérifié que la machine n'est pas en cours d'exécution ou que le nœud n'est pas prêt.
- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez effectué une sauvegarde etcd.



IMPORTANT

Vous devez effectuer une sauvegarde d'etcd avant d'exécuter cette procédure afin de pouvoir restaurer votre cluster en cas de problème.

Procédure

1. Vérifiez et supprimez le membre malsain.

   a. Choisissez un pod qui est *not* sur le nœud affecté :
   Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd -o wide
```

Exemple de sortie

```
etcd-openshift-control-plane-0  5/5  Running  11  3h56m  192.168.10.9  openshift-
control-plane-0  <none>          <none>
etcd-openshift-control-plane-1  5/5  Running  0   3h54m  192.168.10.10  openshift-
```

```
control-plane-1   <none>          <none>
etcd-openshift-control-plane-2  5/5  Running  0  3h58m  192.168.10.11  openshift-
control-plane-2   <none>          <none>
```

b. Se connecter au conteneur etcd en cours d'exécution, en passant le nom d'un pod qui n'est pas sur le nœud affecté :
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

c. Consulter la liste des membres :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+------------------+---------+--------------------+--------------------------+--------------------------
-+---------------------+
| ID               | STATUS  | NAME               | PEER ADDRS               | CLIENT
ADDRS                | IS LEARNER |
+------------------+---------+--------------------+--------------------------+--------------------------
-+---------------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ |
https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ |
https://192.168.10.10:2379/ | false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380/ |
https://192.168.10.9:2379/   | false |
+------------------+---------+--------------------+--------------------------+--------------------------
-+---------------------+
```

Notez l'ID et le nom du membre etcd malsain, car ces valeurs seront nécessaires plus tard dans la procédure. La commande **etcdctl endpoint health** listera le membre supprimé jusqu'à ce que la procédure de remplacement soit terminée et que le nouveau membre soit ajouté.

d. Supprimez le membre etcd malsain en fournissant l'ID à la commande **etcdctl member remove**:

AVERTISSEMENT

Veillez à supprimer le bon membre etcd ; la suppression d'un bon membre etcd peut entraîner une perte de quorum.

```
sh-4.2# etcdctl member remove 7a8197040a5126c8
```

Exemple de sortie

```
Member 7a8197040a5126c8 removed from cluster b23536c33f2cdd1b
```

e. Consultez à nouveau la liste des membres et vérifiez que le membre a bien été supprimé :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+-----------------+---------+-------------------+-------------------------+-------------------------
-+-------------------------+
| ID              | STATUS  | NAME              | PEER ADDRS              | CLIENT
ADDRS              | IS LEARNER |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+-------------------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ |
https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ |
https://192.168.10.10:2379/ | false |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+-------------------------+
```

Vous pouvez maintenant quitter le shell du nœud.

IMPORTANT

Après avoir supprimé le membre, le cluster peut être inaccessible pendant une courte période, le temps que les instances etcd restantes redémarrent.

2. Désactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

Cette commande permet de s'assurer que vous pouvez recréer les secrets et déployer les pods statiques avec succès.

3. Supprimez les anciens secrets du membre etcd malsain qui a été supprimé en exécutant les commandes suivantes.

a. Liste les secrets du membre etcd malsain qui a été supprimé.

```
$ oc get secrets -n openshift-etcd | grep openshift-control-plane-2
```

Saisissez le nom du membre etcd malsain dont vous avez pris note plus tôt dans cette procédure.

Il y a un pair, un serveur et un secret de métrique, comme le montre la sortie suivante :

```
etcd-peer-openshift-control-plane-2              kubernetes.io/tls   2   134m
etcd-serving-metrics-openshift-control-plane-2  kubernetes.io/tls   2   134m
etcd-serving-openshift-control-plane-2           kubernetes.io/tls   2   134m
```

b. Supprime les secrets du membre etcd malsain qui a été supprimé.

i. Supprimer le secret de l'homologue :

```
$ oc delete secret etcd-peer-openshift-control-plane-2 -n openshift-etcd

secret "etcd-peer-openshift-control-plane-2" deleted
```

ii. Supprimer le secret de service :

```
$ oc delete secret etcd-serving-metrics-openshift-control-plane-2 -n openshift-etcd

secret "etcd-serving-metrics-openshift-control-plane-2" deleted
```

iii. Supprimer le secret des métriques :

```
$ oc delete secret etcd-serving-openshift-control-plane-2 -n openshift-etcd

secret "etcd-serving-openshift-control-plane-2" deleted
```

4. Supprimer la machine du plan de contrôle.
Si vous exécutez une infrastructure fournie par l'installateur ou si vous avez utilisé l'API Machine pour créer vos machines, suivez ces étapes. Sinon, vous devez créer le nouveau nœud de plan de contrôle en utilisant la même méthode que celle utilisée pour le créer à l'origine.

a. Obtenir la machine pour le membre en mauvaise santé.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                              STATE
examplecluster-control-plane-0    Running                          3h11m   openshift-control-
plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
3ff2-41c5-b099-0aa41222964e   externally provisioned (1)
examplecluster-control-plane-1    Running                          3h11m   openshift-control-
plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
329c-475e-8d81-03b20280a3e1   externally provisioned
examplecluster-control-plane-2    Running                          3h11m   openshift-control-
plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
61d8-410f-be5b-6a395b056135   externally provisioned
examplecluster-compute-0          Running                          165m    openshift-compute-0
baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
13a31858241f          provisioned
examplecluster-compute-1          Running                          165m    openshift-compute-1
baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
e7ea72ab13b9          provisioned
```

(1) Il s'agit de la machine du plan de contrôle pour le nœud malsain, **examplecluster-control-plane-2**.

b. Enregistrez la configuration de la machine dans un fichier sur votre système de fichiers :

```
$ oc get machine examplecluster-control-plane-2 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 Indiquez le nom de la machine du plan de contrôle pour le nœud malsain.

c. Modifiez le fichier **new-master-machine.yaml** créé à l'étape précédente pour lui attribuer un nouveau nom et supprimer les champs inutiles.

i. Retirer toute la section **status**:

```
status:
  addresses:
  - address: ""
    type: InternalIP
  - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.373
    type: InternalDNS
  - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.371
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Machine
    name: fe80::4adf:37ff:feb0:8aa1%ens1f1.372
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: machine.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: Machine
```

5. Changez le nom du champ **metadata.name**.
Il est recommandé de conserver le même nom de base que l'ancienne machine et de remplacer le numéro de fin par le prochain numéro disponible. Dans cet exemple, **examplecluster-control-plane-2** est remplacé par **examplecluster-control-plane-3**.

Par exemple :

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: examplecluster-control-plane-3
  ...
```

a. Supprimer le champ **spec.providerID**:

```
  providerID: baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-61d8-410f-be5b-6a395b056135
```

b. Supprimer les champs **metadata.annotations** et **metadata.generation**:

```
  annotations:
    machine.openshift.io/instance-state: externally provisioned
  ...
  generation: 2
```

c. Supprimer les champs **spec.conditions**, **spec.lastUpdated**, **spec.nodeRef** et **spec.phase**:

```
    lastTransitionTime: "2022-08-03T08:40:36Z"
  message: 'Drain operation currently blocked by: [{Name:EtcdQuorumOperator
  Owner:clusteroperator/etcd}]'
  reason: HookPresent
  severity: Warning
  status: "False"

  type: Drainable
  lastTransitionTime: "2022-08-03T08:39:55Z"
  status: "True"
  type: InstanceExists

  lastTransitionTime: "2022-08-03T08:36:37Z"
  status: "True"
  type: Terminable
  lastUpdated: "2022-08-03T08:40:36Z"
  nodeRef:
  kind: Node
  name: openshift-control-plane-2
  uid: 788df282-6507-4ea2-9a43-24f237ccbc3c
  phase: Running
```

6. Assurez-vous que l'opérateur Bare Metal est disponible en exécutant la commande suivante :

```
$ oc get clusteroperator baremetal
```

Exemple de sortie

```
NAME      VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
baremetal 4.12.0    True        False         False      3d15h
```

7. Supprimez l'ancien objet **BareMetalHost** en exécutant la commande suivante :

```
$ oc delete bmh openshift-control-plane-2 -n openshift-machine-api
```

Exemple de sortie

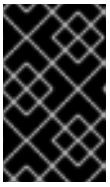
```
baremetalhost.metal3.io "openshift-control-plane-2" deleted
```

8. Supprimez la machine du membre malsain en exécutant la commande suivante :

```
$ oc delete machine -n openshift-machine-api examplecluster-control-plane-2
```

Après avoir supprimé les objets **BareMetalHost** et **Machine**, le contrôleur **Machine** supprime automatiquement l'objet **Node**.

Si la suppression de la machine est retardée pour une raison quelconque ou si la commande est entravée et retardée, vous pouvez forcer la suppression en supprimant le champ finalizer de l'objet machine.



IMPORTANT

N'interrompez pas l'effacement de la machine en appuyant sur **Ctrl c**. Vous devez laisser la commande se dérouler jusqu'à son terme. Ouvrez une nouvelle fenêtre de terminal pour éditer et supprimer les champs du finalisateur.

a. Modifiez la configuration de la machine en exécutant la commande suivante :

```
$ oc edit machine -n openshift-machine-api examplecluster-control-plane-2
```

b. Supprimez les champs suivants dans la ressource personnalisée **Machine**, puis enregistrez le fichier mis à jour :

```
finalizers:
- machine.machine.openshift.io
```

Exemple de sortie

```
machine.machine.openshift.io/examplecluster-control-plane-2 edited
```

9. Vérifiez que la machine a été supprimée en exécutant la commande suivante :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                              STATE
examplecluster-control-plane-0    Running                          3h11m   openshift-control-plane-0
baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-3ff2-41c5-b099-
0aa41222964e   externally provisioned
examplecluster-control-plane-1    Running                          3h11m   openshift-control-plane-1
baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-329c-475e-8d81-
03b20280a3e1   externally provisioned
examplecluster-compute-0          Running                          165m    openshift-compute-0
baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
13a31858241f          provisioned
examplecluster-compute-1          Running                          165m    openshift-compute-1
baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
e7ea72ab13b9          provisioned
```

10. Vérifiez que le nœud a été supprimé en exécutant la commande suivante :

```
$ oc get nodes
```

```
NAME                    STATUS ROLES   AGE   VERSION
openshift-control-plane-0 Ready master 3h24m v1.25.0
openshift-control-plane-1 Ready master 3h24m v1.25.0
openshift-compute-0       Ready worker 176m v1.25.0
openshift-compute-1       Ready worker 176m v1.25.0
```

11. Créez le nouvel objet **BareMetalHost** et le secret pour stocker les informations d'identification BMC :

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openshift-control-plane-2-bmc-secret
  namespace: openshift-machine-api
data:
  password: <password>
  username: <username>
type: Opaque
---
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
metadata:
  name: openshift-control-plane-2
  namespace: openshift-machine-api
spec:
  automatedCleaningMode: disabled
  bmc:
    address: redfish://10.46.61.18:443/redfish/v1/Systems/1
    credentialsName: openshift-control-plane-2-bmc-secret
    disableCertificateVerification: true
  bootMACAddress: 48:df:37:b0:8a:a0
  bootMode: UEFI
  externallyProvisioned: false
  online: true
  rootDeviceHints:
    deviceName: /dev/sda
  userData:
    name: master-user-data-managed
    namespace: openshift-machine-api
EOF
```

NOTE

Le nom d'utilisateur et le mot de passe peuvent être trouvés dans les secrets de l'autre hôte bare metal. Le protocole à utiliser dans **bmc:address** peut être obtenu à partir d'autres objets bmh.

IMPORTANT

Si vous réutilisez la définition de l'objet **BareMetalHost** à partir d'un hôte de plan de contrôle existant, ne laissez pas le champ **externallyProvisioned** sur **true**.

Les objets du plan de contrôle **BareMetalHost** existants peuvent avoir l'indicateur **externallyProvisioned** défini sur **true** s'ils ont été provisionnés par le programme d'installation d'OpenShift Container Platform.

Une fois l'inspection terminée, l'objet **BareMetalHost** est créé et disponible pour être approvisionné.

12. Vérifier le processus de création à l'aide des objets disponibles sur **BareMetalHost**:

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

NAME                 STATE                  CONSUMER                       ONLINE ERROR   AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 available          examplecluster-control-plane-3 true        47m
openshift-compute-0       provisioned        examplecluster-compute-0       true        4h48m
openshift-compute-1       provisioned        examplecluster-compute-1       true        4h48m
```

a. Créez la nouvelle machine du plan de contrôle à l'aide du fichier **new-master-machine.yaml**:

```
$ oc apply -f new-master-machine.yaml
```

b. Vérifiez que la nouvelle machine a été créée :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie

```
NAME                                 PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                                 STATE
examplecluster-control-plane-0       Running                          3h11m   openshift-control-
plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
3ff2-41c5-b099-0aa41222964e   externally provisioned (1)
examplecluster-control-plane-1       Running                          3h11m   openshift-control-
plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
329c-475e-8d81-03b20280a3e1   externally provisioned
examplecluster-control-plane-2       Running                          3h11m   openshift-control-
plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
61d8-410f-be5b-6a395b056135   externally provisioned
examplecluster-compute-0             Running                          165m    openshift-compute-
0         baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-
4bb3-80ec-13a31858241f        provisioned
examplecluster-compute-1             Running                          165m    openshift-compute-
1         baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-
4241-91dc-e7ea72ab13b9        provisioned
```

1 La nouvelle machine, **clustername-8qw5l-master-3**, est en cours de création et sera prête après le changement de phase de **Provisioning** à **Running**.

La création de la nouvelle machine devrait prendre quelques minutes. L'opérateur du cluster etcd se synchronisera automatiquement lorsque la machine ou le nœud reviendra à un état sain.

c. Vérifiez que l'hôte bare metal est provisionné et qu'aucune erreur n'est signalée en exécutant la commande suivante :

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

Exemple de sortie

```
$ oc get bmh -n openshift-machine-api
NAME                 STATE                CONSUMER                    ONLINE ERROR AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 provisioned          examplecluster-control-plane-3 true
47m
openshift-compute-0      provisioned          examplecluster-compute-0      true
4h48m
openshift-compute-1      provisioned          examplecluster-compute-1      true
4h48m
```

d. Vérifiez que le nouveau nœud est ajouté et prêt à fonctionner en exécutant la commande suivante :

```
$ oc get nodes
```

Exemple de sortie

```
$ oc get nodes
NAME                 STATUS ROLES   AGE   VERSION
openshift-control-plane-0 Ready master 4h26m v1.25.0
openshift-control-plane-1 Ready master 4h26m v1.25.0
openshift-control-plane-2 Ready master 12m   v1.25.0
openshift-compute-0      Ready worker 3h58m v1.25.0
openshift-compute-1      Ready worker 3h58m v1.25.0
```

13. Réactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

14. Vous pouvez vérifier que la section **unsupportedConfigOverrides** est supprimée de l'objet en entrant cette commande :

```
$ oc get etcd/cluster -oyaml
```

15. Si vous utilisez OpenShift à nœud unique, redémarrez le nœud. Sinon, vous risquez de rencontrer l'erreur suivante dans l'opérateur de cluster etcd :

Exemple de sortie

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

Vérification

1. Vérifiez que tous les pods etcd fonctionnent correctement.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

Exemple de sortie

```
etcd-openshift-control-plane-0     5/5     Running     0     105m
etcd-openshift-control-plane-1     5/5     Running     0     107m
etcd-openshift-control-plane-2     5/5     Running     0     103m
```

Si la sortie de la commande précédente n'indique que deux pods, vous pouvez forcer manuellement un redéploiement d'etcd. Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc patch etcd cluster -p='{"spec" : {\i1}"forceRedeploymentReason" : \N-"recovery-'\N"$(
date --rfc-3339=ns )'\N'\N"}" -type=merge} --type=merge (1)
```

(1) La valeur **forceRedeploymentReason** doit être unique, c'est pourquoi un horodatage est ajouté.

Pour vérifier qu'il y a exactement trois membres etcd, connectez-vous au conteneur etcd en cours d'exécution, en indiquant le nom d'un pod qui n'était pas sur le nœud affecté. Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

2. Consulter la liste des membres :

```
sh-4.2# etcdctl member list -w table
```

Exemple de sortie

```
+------------------+---------+--------------------+--------------------------+--------------------------+---
--------------+
|        ID        | STATUS  |        NAME        |        PEER ADDRS        |        CLIENT ADDRS
```

```
|    IS LEARNER    |
+-----------------+---------+-------------------+-------------------------+-------------------------+---
-------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380 |
https://192.168.10.11:2379 |    false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380 |
https://192.168.10.10:2379 |    false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380 |
https://192.168.10.9:2379 |     false |
+-----------------+---------+-------------------+-------------------------+-------------------------+---
-------------+
```

NOTE

Si la sortie de la commande précédente répertorie plus de trois membres etcd, vous devez supprimer avec précaution le membre indésirable.

3. Vérifiez que tous les membres d'etcd sont sains en exécutant la commande suivante :

```
# etcdctl endpoint health --cluster
```

Exemple de sortie

```
https://192.168.10.10:2379 is healthy: successfully committed proposal: took = 8.973065ms
https://192.168.10.9:2379 is healthy: successfully committed proposal: took = 11.559829ms
https://192.168.10.11:2379 is healthy: successfully committed proposal: took = 11.665203ms
```

4. Validez que tous les nœuds sont à la dernière révision en exécutant la commande suivante :

```
$ oc get etcd -o=jsonpath='{range.items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

```
AllNodesAtLatestRevision
```

5.2.5. Ressources complémentaires

- Protection du quorum à l'aide de crochets de cycle de vie de la machine

5.3. SAUVEGARDE ET RESTAURATION D'ETCD SUR UN CLUSTER HÉBERGÉ

Si vous utilisez des plans de contrôle hébergés sur OpenShift Container Platform, le processus de sauvegarde et de restauration d'etcd est différent du processus de sauvegarde habituel d'etcd.

IMPORTANT

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

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

5.3.1. Prendre un instantané d'etcd sur un cluster hébergé

Dans le cadre du processus de sauvegarde d'etcd pour un cluster hébergé, vous prenez un instantané d'etcd. Après avoir pris l'instantané, vous pouvez le restaurer, par exemple, dans le cadre d'une opération de reprise après sinistre.

IMPORTANT

Cette procédure nécessite un temps d'arrêt de l'API.

Procédure

1. Interrompez la réconciliation du cluster hébergé en entrant cette commande :

```
$ oc patch -n clusters hostedclusters/${CLUSTER_NAME} -p '{"spec":
{"pausedUntil":"${PAUSED_UNTIL}"}}' --type=merge
```

2. Arrêtez tous les déploiements d'etcd-writer en entrant cette commande :

```
$ oc scale deployment -n ${HOSTED_CLUSTER_NAMESPACE} --replicas=0 kube-
apiserver openshift-apiserver openshift-oauth-apiserver
```

3. Prenez un instantané etcd en utilisant la commande **exec** dans chaque conteneur etcd :

```
$ oc exec -it etcd-0 -n ${HOSTED_CLUSTER_NAMESPACE} -- env ETCDCTL_API=3
/usr/bin/etcdctl --cacert /etc/etcd/tls/client/etcd-client-ca.crt --cert /etc/etcd/tls/client/etcd-
client.crt --key /etc/etcd/tls/client/etcd-client.key --endpoints=localhost:2379 snapshot save
/var/lib/data/snapshot.db
$ oc exec -it etcd-0 -n ${HOSTED_CLUSTER_NAMESPACE} -- env ETCDCTL_API=3
/usr/bin/etcdctl -w table snapshot status /var/lib/data/snapshot.db
```

4. Copiez les données de l'instantané vers un emplacement où vous pourrez les récupérer ultérieurement, tel qu'un seau S3, comme indiqué dans l'exemple suivant.

NOTE

L'exemple suivant utilise la version 2 de la signature. Si vous vous trouvez dans une région qui prend en charge la version 4 de la signature, comme la région us-east-2, utilisez la version 4 de la signature. Dans le cas contraire, si vous utilisez la version 2 de la signature pour copier l'instantané dans un seau S3, le téléchargement échoue et la version 2 de la signature est obsolète.

Exemple

```
BUCKET_NAME=somebucket
FILEPATH="/${BUCKET_NAME}/${CLUSTER_NAME}-snapshot.db"
CONTENT_TYPE="application/x-compressed-tar"
DATE_VALUE=`date -R`
SIGNATURE_STRING="PUT\n\n${CONTENT_TYPE}\n${DATE_VALUE}\n${FILEPATH}"
ACCESS_KEY=accesskey
SECRET_KEY=secret
SIGNATURE_HASH=`echo -en ${SIGNATURE_STRING} | openssl sha1 -hmac
${SECRET_KEY} -binary | base64`

oc exec -it etcd-0 -n ${HOSTED_CLUSTER_NAMESPACE} -- curl -X PUT -T
"/var/lib/data/snapshot.db" \
  -H "Host: ${BUCKET_NAME}.s3.amazonaws.com" \
  -H "Date: ${DATE_VALUE}" \
  -H "Content-Type: ${CONTENT_TYPE}" \
  -H "Authorization: AWS ${ACCESS_KEY}:${SIGNATURE_HASH}" \
  https://${BUCKET_NAME}.s3.amazonaws.com/${CLUSTER_NAME}-snapshot.db
```

5. Si vous souhaitez pouvoir restaurer ultérieurement l'instantané sur un nouveau cluster, enregistrez le secret de chiffrement auquel le cluster hébergé fait référence, comme indiqué dans cet exemple :

Exemple

```
oc get hostedcluster $CLUSTER_NAME -o=jsonpath='{.spec.secretEncryption.aescbc}'
{"activeKey":{"name":"CLUSTER_NAME-etcd-encryption-key"}}

# Save this secret, or the key it contains so the etcd data can later be decrypted
oc get secret ${CLUSTER_NAME}-etcd-encryption-key -o=jsonpath='{.data.key}'
```

Prochaines étapes

Restaurer l'instantané etcd.

5.3.2. Restauration d'un snapshot etcd sur un cluster hébergé

Si vous avez un instantané d'etcd de votre cluster hébergé, vous pouvez le restaurer. Actuellement, vous ne pouvez restaurer un instantané etcd que lors de la création d'un cluster.

Pour restaurer un instantané etcd, vous modifiez la sortie de la commande **create cluster --render** et définissez une valeur **restoreSnapshotURL** dans la section etcd de la spécification **HostedCluster**.

Conditions préalables

Vous avez pris un instantané etcd sur un cluster hébergé.

Procédure

1. Sur l'interface de ligne de commande (CLI) **aws**, créez une URL pré-signée afin de pouvoir télécharger votre instantané etcd depuis S3 sans transmettre d'informations d'identification au déploiement etcd :

```
ETCD_SNAPSHOT=${ETCD_SNAPSHOT:-"s3://${BUCKET_NAME}/${CLUSTER_NAME}-snapshot.db"}
ETCD_SNAPSHOT_URL=$(aws s3 presign ${ETCD_SNAPSHOT})
```

2. Modifier la spécification **HostedCluster** pour faire référence à l'URL :

```
spec:
  etcd:
    managed:
      storage:
        persistentVolume:
          size: 4Gi
        type: PersistentVolume
        restoreSnapshotURL:
        - "${ETCD_SNAPSHOT_URL}"
    managementType: Managed
```

3. Assurez-vous que le secret que vous avez référencé à partir de la valeur **spec.secretEncryption.aescbc** contient la même clé AES que celle que vous avez enregistrée dans les étapes précédentes.

5.3.3. Ressources complémentaires

- Reprise après sinistre pour un cluster hébergé dans une région AWS

5.4. REPRISE APRÈS SINISTRE

5.4.1. À propos de la reprise après sinistre

La documentation sur la reprise après sinistre fournit des informations aux administrateurs sur la façon de reprendre après plusieurs situations de sinistre pouvant survenir avec leur cluster OpenShift Container Platform. En tant qu'administrateur, vous devrez peut-être suivre une ou plusieurs des procédures suivantes pour remettre votre cluster en état de fonctionnement.

IMPORTANT

La reprise après sinistre nécessite la présence d'au moins un hôte de plan de contrôle sain.

Rétablissement d'un état antérieur de la grappe

Cette solution gère les situations où vous souhaitez restaurer votre cluster à un état antérieur, par exemple, si un administrateur supprime quelque chose de critique. Cela inclut également les situations où vous avez perdu la majorité de vos hôtes du plan de contrôle, ce qui entraîne la perte du quorum etcd et la mise hors ligne du cluster. Tant que vous avez effectué une sauvegarde etcd, vous pouvez suivre cette procédure pour restaurer votre cluster dans un état antérieur.
Le cas échéant, il peut également s'avérer nécessaire de récupérer des certificats de plan de contrôle expirés.

AVERTISSEMENT

La restauration d'un état antérieur de la grappe est une action destructrice et déstabilisante pour une grappe en cours d'exécution. Cette procédure ne doit être utilisée qu'en dernier recours.

Avant d'effectuer une restauration, voir À propos de la restauration de l'état du cluster pour plus d'informations sur l'impact sur le cluster.

NOTE

Si la majorité de vos maîtres sont encore disponibles et que vous avez un quorum etcd, suivez la procédure pour remplacer un seul membre etcd en mauvaise santé .

Récupération des certificats expirés du plan de contrôle

Cette solution permet de gérer les situations où les certificats du plan de contrôle ont expiré. Par exemple, si vous arrêtez votre cluster avant la première rotation des certificats, qui a lieu 24 heures après l'installation, vos certificats ne seront pas renouvelés et expireront. Vous pouvez suivre cette procédure pour récupérer les certificats de plan de contrôle expirés.

5.4.2. Rétablissement d'un état antérieur de la grappe

Pour restaurer le cluster à un état antérieur, vous devez avoir préalablement sauvegardé les données etcd en créant un snapshot. Vous utiliserez cet instantané pour restaurer l'état du cluster.

5.4.2.1. À propos de la restauration de l'état des clusters

Vous pouvez utiliser une sauvegarde etcd pour restaurer votre cluster à un état antérieur. Cela peut être utilisé pour récupérer les situations suivantes :

- Le cluster a perdu la majorité des hôtes du plan de contrôle (perte de quorum).
- Un administrateur a supprimé un élément critique et doit restaurer le cluster.

AVERTISSEMENT

La restauration d'un état antérieur de la grappe est une action destructrice et déstabilisante pour une grappe en cours d'exécution. Elle ne doit être utilisée qu'en dernier recours.

Si vous êtes en mesure de récupérer des données à l'aide du serveur API Kubernetes, alors etcd est disponible et vous ne devez pas restaurer à l'aide d'une sauvegarde etcd.

La restauration d'etcd ramène effectivement un cluster dans le temps et tous les clients connaîtront un

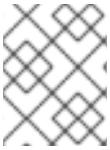
historique parallèle et conflictuel. Cela peut avoir un impact sur le comportement des composants de surveillance tels que les kubelets, les gestionnaires de contrôleurs Kubernetes, les contrôleurs SDN et les contrôleurs de volumes persistants.

Les opérateurs du serveur API Kubernetes, du gestionnaire de contrôleur Kubernetes, du planificateur Kubernetes et de etcd peuvent se retrouver bloqués lorsque les fichiers sur le disque sont en conflit avec le contenu de etcd. Cela peut nécessiter des actions manuelles pour résoudre les problèmes.

Dans les cas extrêmes, le cluster peut perdre la trace des volumes persistants, supprimer des charges de travail critiques qui n'existent plus, réimager des machines et réécrire des bundles d'autorité de certification avec des certificats expirés.

5.4.2.2. Rétablissement d'un état antérieur de la grappe

Vous pouvez utiliser une sauvegarde etcd enregistrée pour restaurer un état antérieur du cluster ou restaurer un cluster qui a perdu la majorité des hôtes du plan de contrôle.



NOTE

Si votre cluster utilise un jeu de machines de plan de contrôle, voir "Dépannage du jeu de machines de plan de contrôle" pour une procédure de récupération etcd plus simple.

IMPORTANT

Lorsque vous restaurez votre cluster, vous devez utiliser une sauvegarde etcd provenant de la même version de z-stream. Par exemple, un cluster OpenShift Container Platform 4.7.2 doit utiliser une sauvegarde etcd provenant de la version 4.7.2.

Conditions préalables

- Accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Un hôte de plan de contrôle sain à utiliser comme hôte de reprise.
- Accès SSH aux hôtes du plan de contrôle.
- Un répertoire de sauvegarde contenant à la fois l'instantané etcd et les ressources pour les pods statiques, qui proviennent de la même sauvegarde. Les noms de fichiers dans le répertoire doivent être dans les formats suivants : **snapshot_<datetimestamp>.db** et **static_kuberesources_<datetimestamp>.tar.gz**.

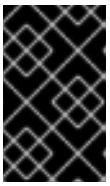
IMPORTANT

Pour les nœuds de plan de contrôle sans récupération, il n'est pas nécessaire d'établir une connectivité SSH ou d'arrêter les pods statiques. Vous pouvez supprimer et recréer d'autres machines de plan de contrôle sans récupération, une par une.

Procédure

1. Sélectionnez un hôte du plan de contrôle à utiliser comme hôte de restauration. Il s'agit de l'hôte sur lequel vous exécuterez l'opération de restauration.
2. Établir une connectivité SSH avec chacun des nœuds du plan de contrôle, y compris l'hôte de reprise.
   Le serveur API Kubernetes devient inaccessible après le démarrage du processus de

restauration, de sorte que vous ne pouvez pas accéder aux nœuds du plan de contrôle. Pour cette raison, il est recommandé d'établir une connectivité SSH à chaque hôte du plan de contrôle dans un terminal séparé.

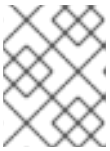


IMPORTANT

Si vous n'effectuez pas cette étape, vous ne pourrez pas accéder aux hôtes du plan de contrôle pour terminer la procédure de restauration, et vous ne pourrez pas récupérer votre cluster à partir de cet état.

3. Copiez le répertoire de sauvegarde etcd sur l'hôte du plan de contrôle de reprise.
Cette procédure suppose que vous avez copié le répertoire **backup** contenant le snapshot etcd et les ressources pour les pods statiques dans le répertoire /**home**/**core**/ de votre hôte de plan de contrôle de récupération.

4. Arrêtez les pods statiques sur tous les autres nœuds du plan de contrôle.



NOTE

Il n'est pas nécessaire d'arrêter manuellement les pods sur l'hôte de récupération. Le script de récupération arrêtera les pods sur l'hôte de récupération.

a. Accéder à un hôte du plan de contrôle qui n'est pas l'hôte de reprise.

b. Déplacer le fichier pod etcd existant hors du répertoire kubelet manifest :

```
$ sudo mv /etc/kubernetes/manifests/etcd-pod.yaml /tmp
```

c. Vérifiez que les pods etcd sont arrêtés.

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

La sortie de cette commande doit être vide. Si ce n'est pas le cas, attendez quelques minutes et vérifiez à nouveau.

d. Déplacez le fichier pod du serveur API Kubernetes existant hors du répertoire kubelet manifest :

```
$ sudo mv /etc/kubernetes/manifests/kube-apiserver-pod.yaml /tmp
```

e. Vérifiez que les pods du serveur API Kubernetes sont arrêtés.

```
$ sudo crictl ps | grep kube-apiserver | egrep -v "operator|guard"
```

La sortie de cette commande doit être vide. Si ce n'est pas le cas, attendez quelques minutes et vérifiez à nouveau.

f. Déplacez le répertoire de données etcd vers un autre emplacement :

```
$ sudo mv /var/lib/etcd/ /tmp
```

g. Répétez cette étape sur chacun des autres hôtes du plan de contrôle qui n'est pas l'hôte de reprise.

5. Accéder à l'hôte du plan de contrôle de récupération.

6. Si le proxy à l'échelle du cluster est activé, assurez-vous que vous avez exporté les variables d'environnement **NO_PROXY**, **HTTP_PROXY**, et **HTTPS_PROXY**.

   ASTUCE

   Vous pouvez vérifier si le proxy est activé en examinant la sortie de **oc get proxy cluster -o yaml**. Le proxy est activé si les champs **httpProxy**, **httpsProxy** et **noProxy** ont des valeurs définies.

7. Exécutez le script de restauration sur l'hôte du plan de contrôle de récupération et indiquez le chemin d'accès au répertoire de sauvegarde etcd :

   ```
   $ sudo -E /usr/local/bin/cluster-restore.sh /home/core/backup
   ```

   Exemple de sortie de script

   ```
   ...stopping kube-scheduler-pod.yaml
   ...stopping kube-controller-manager-pod.yaml
   ...stopping etcd-pod.yaml
   ...stopping kube-apiserver-pod.yaml
   Waiting for container etcd to stop
   .complete
   Waiting for container etcdctl to stop
   .....complete
   Waiting for container etcd-metrics to stop
   complete
   Waiting for container kube-controller-manager to stop
   complete
   Waiting for container kube-apiserver to stop
   .....complete
   Waiting for container kube-scheduler to stop
   complete
   Moving etcd data-dir /var/lib/etcd/member to /var/lib/etcd-backup
   starting restore-etcd static pod
   starting kube-apiserver-pod.yaml
   static-pod-resources/kube-apiserver-pod-7/kube-apiserver-pod.yaml
   starting kube-controller-manager-pod.yaml
   static-pod-resources/kube-controller-manager-pod-7/kube-controller-manager-pod.yaml
   starting kube-scheduler-pod.yaml
   static-pod-resources/kube-scheduler-pod-8/kube-scheduler-pod.yaml
   ```

   NOTE

   Le processus de restauration peut entraîner l'entrée des nœuds dans l'état **NotReady** si les certificats des nœuds ont été mis à jour après la dernière sauvegarde etcd.

8. Vérifiez que les nœuds sont dans l'état **Ready**.

   a. Exécutez la commande suivante :

      ```
      $ oc get nodes -w
      ```

Exemple de sortie

```
NAME              STATUS  ROLES         AGE     VERSION
host-172-25-75-28   Ready   master        3d20h   v1.25.0
host-172-25-75-38   Ready   infra,worker  3d20h   v1.25.0
host-172-25-75-40   Ready   master        3d20h   v1.25.0
host-172-25-75-65   Ready   master        3d20h   v1.25.0
host-172-25-75-74   Ready   infra,worker  3d20h   v1.25.0
host-172-25-75-79   Ready   worker        3d20h   v1.25.0
host-172-25-75-86   Ready   worker        3d20h   v1.25.0
host-172-25-75-98   Ready   infra,worker  3d20h   v1.25.0
```

Il peut s'écouler plusieurs minutes avant que tous les nœuds ne communiquent leur état.

b. Si des nœuds sont dans l'état **NotReady**, connectez-vous aux nœuds et supprimez tous les fichiers PEM du répertoire **/var/lib/kubelet/pki** sur chaque nœud. Vous pouvez vous connecter aux nœuds par SSH ou utiliser la fenêtre de terminal de la console web.

```
$  ssh -i <ssh-key-path> core@<master-hostname>
```

Exemple de répertoire pki

```
sh-4.4# pwd
/var/lib/kubelet/pki
sh-4.4# ls
kubelet-client-2022-04-28-11-24-09.pem  kubelet-server-2022-04-28-11-24-15.pem
kubelet-client-current.pem              kubelet-server-current.pem
```

9. Redémarrer le service kubelet sur tous les hôtes du plan de contrôle.

a. À partir de l'hôte de récupération, exécutez la commande suivante :

```
$ sudo systemctl restart kubelet.service
```

b. Répétez cette étape sur tous les autres hôtes du plan de contrôle.

10. Approuver les RSC en attente :

a. Obtenir la liste des CSR actuels :

```
$ oc get csr
```

Exemple de sortie

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2s94x   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending 1
csr-4bd6t   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending 2
csr-4hl85   13m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

```
3
csr-zhhhp   3m8s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
4
...
```

1 2 Un CSR de service kubelet en attente (pour les installations fournies par l'utilisateur).

3 4 Un CSR **node-bootstrapper** en attente.

b. Examinez les détails d'un CSR pour vérifier qu'il est valide :

```
oc describe csr <csr_name> 1
```

1 **<csr_name>** est le nom d'un CSR figurant dans la liste des CSR actuels.

c. Approuver chaque CSR **node-bootstrapper** valide :

```
$ oc adm certificate approve <csr_name>
```

d. Pour les installations fournies par l'utilisateur, approuver chaque CSR de service kubelet valide :

```
$ oc adm certificate approve <csr_name>
```

11. Vérifiez que le plan de contrôle à membre unique a bien démarré.

a. Depuis l'hôte de récupération, vérifiez que le conteneur etcd est en cours d'exécution.

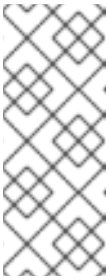
```
$ sudo crictl ps | grep etcd | grep -v operator
```

Exemple de sortie

```
3ad41b7908e32
36f86e2eeaaffe662df0d21041eb22b8198e0e58abeeae8c743c3e6e977e8009
About a minute ago   Running             etcd                                    0
7c05f8af362f0
```

b. Depuis l'hôte de récupération, vérifiez que le pod etcd est en cours d'exécution.

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```



NOTE

Si vous essayez d'exécuter **oc login** avant d'exécuter cette commande et que vous recevez l'erreur suivante, attendez quelques instants pour que les contrôleurs d'authentification démarrent et réessayez.

```
Unable to connect to the server: EOF
```

Exemple de sortie

```
NAME                                 READY   STATUS      RESTARTS   AGE
etcd-ip-10-0-143-125.ec2.internal    1/1     Running     1          2m47s
```

Si l'état est **Pending**, ou si la sortie indique plus d'un pod etcd en cours d'exécution, attendez quelques minutes et vérifiez à nouveau.

NOTE

N'effectuez l'étape suivante que si vous utilisez le plugin réseau **OVNKubernetes**.

12. Supprimer les objets nœuds associés aux hôtes du plan de contrôle qui ne sont pas l'hôte du plan de contrôle de reprise.

```
oc delete node <non-recovery-controlplane-host-1> <non-recovery-controlplane-host-2>
```

13. Vérifier que le Cluster Network Operator (CNO) redéploie le plan de contrôle OVN-Kubernetes et qu'il ne référence plus les mauvaises adresses IP des contrôleurs. Pour vérifier ce résultat, vérifiez régulièrement la sortie de la commande suivante. Attendez qu'elle renvoie un résultat vide avant de passer à l'étape suivante.

```
$ oc -n openshift-ovn-kubernetes get ds/ovnkube-master -o yaml | grep -E
'<wrong_master_ip_1>|<wrong_master_ip_2>'
```

NOTE

Cela peut prendre au moins 5 à 10 minutes pour que le plan de contrôle OVN-Kubernetes soit redéployé et que la commande précédente renvoie une sortie vide.

14. Désactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

Cette commande permet de s'assurer que vous pouvez recréer les secrets et déployer les pods statiques avec succès.

15. Redémarrez les pods Kubernetes d'Open Virtual Network (OVN) sur tous les hôtes.

NOTE

Les webhooks de validation et de mutation des admissions peuvent rejeter les pods. Si vous ajoutez d'autres webhooks dont l'adresse **failurePolicy** est **Fail**, ils peuvent rejeter des pods et le processus de restauration peut échouer. Vous pouvez éviter cela en sauvegardant et en supprimant les webhooks lors de la restauration de l'état de la grappe. Une fois l'état du cluster restauré avec succès, vous pouvez réactiver les webhooks.

Vous pouvez également définir temporairement **failurePolicy** sur **Ignore** pendant la restauration de l'état de la grappe. Une fois l'état de la grappe restauré avec succès, vous pouvez définir **failurePolicy** sur **Fail**.

a. Supprimez la base de données en direction du nord (nbdb) et la base de données en direction du sud (sbdb). Accédez à l'hôte de reprise et aux nœuds de plan de contrôle restants à l'aide de Secure Shell (SSH) et exécutez la commande suivante :

```
$ sudo rm -f /var/lib/ovn/etc/*.db
```

b. Supprimez tous les pods du plan de contrôle OVN-Kubernetes en exécutant la commande suivante :

```
$ oc delete pods -l app=ovnkube-master -n openshift-ovn-kubernetes
```

c. Assurez-vous que tous les pods du plan de contrôle OVN-Kubernetes sont à nouveau déployés et sont dans un état **Running** en exécutant la commande suivante :

```
$ oc get pods -l app=ovnkube-master -n openshift-ovn-kubernetes
```

Exemple de sortie

```
NAME                  READY   STATUS    RESTARTS   AGE
ovnkube-master-nb24h   4/4     Running   0          48s
ovnkube-master-rm8kw   4/4     Running   0          47s
ovnkube-master-zbqnh   4/4     Running   0          56s
```

d. Supprimez tous les pods **ovnkube-node** en exécutant la commande suivante :

```
$ oc get pods -n openshift-ovn-kubernetes -o name | grep ovnkube-node | while read p ;
do oc delete $p -n openshift-ovn-kubernetes ; done
```

e. Assurez-vous que tous les pods **ovnkube-node** sont à nouveau déployés et sont dans un état **Running** en exécutant la commande suivante :

```
$ oc get  pods -n openshift-ovn-kubernetes | grep ovnkube-node
```

16. Supprimer et recréer les autres machines du plan de contrôle qui ne sont pas des machines de récupération, une par une. Une fois les machines recréées, une nouvelle révision est forcée et etcd passe automatiquement à l'échelle supérieure.

- Si vous utilisez une installation bare metal fournie par l'utilisateur, vous pouvez recréer une machine de plan de contrôle en utilisant la même méthode que celle utilisée pour la créer à l'origine. Pour plus d'informations, voir "Installation d'un cluster fourni par l'utilisateur sur

bare metal".

AVERTISSEMENT

Ne supprimez pas et ne recréez pas la machine pour l'hôte de récupération.

- Si vous utilisez une infrastructure fournie par l'installateur ou si vous avez utilisé l'API Machine pour créer vos machines, procédez comme suit :

AVERTISSEMENT

Ne supprimez pas et ne recréez pas la machine pour l'hôte de récupération.

Pour les installations bare metal sur une infrastructure fournie par l'installateur, les machines du plan de contrôle ne sont pas recréées. Pour plus d'informations, voir "Remplacement d'un nœud de plan de contrôle bare-metal".

a. Obtenir la machine de l'un des hôtes du plan de contrôle perdus.
Dans un terminal ayant accès au cluster en tant qu'utilisateur cluster-admin, exécutez la commande suivante :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie :

```
NAME                                PHASE   TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-0          Running m4.xlarge   us-east-1   us-east-1a
3h37m   ip-10-0-131-183.ec2.internal   aws:///us-east-1a/i-0ec2782f8287dfb7e
stopped (1)
clustername-8qw5l-master-1          Running m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631
running
clustername-8qw5l-master-2          Running m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba
running
clustername-8qw5l-worker-us-east-1a-wbtgd  Running m4.large   us-east-1   us-
east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-
010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running m4.large   us-east-1   us-
east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-
0cb45ac45a166173b   running
```

```
clustername-8qw5l-worker-us-east-1c-pkg26  Running  m4.large  us-east-1  us-east-1c  3h28m  ip-10-0-170-181.ec2.internal  aws:///us-east-1c/i-06861c00007751b0a  running
```

1. Il s'agit de la machine du plan de contrôle de l'hôte du plan de contrôle perdu, **ip-10-0-131-183.ec2.internal**.

b. Enregistrez la configuration de la machine dans un fichier sur votre système de fichiers :

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1. Indiquez le nom de la machine du plan de contrôle pour l'hôte du plan de contrôle perdu.

c. Modifiez le fichier **new-master-machine.yaml** créé à l'étape précédente pour lui attribuer un nouveau nom et supprimer les champs inutiles.

i. Retirer toute la section **status**:

```
status:
  addresses:
  - address: 10.0.131.183
    type: InternalIP
  - address: ip-10-0-131-183.ec2.internal
    type: InternalDNS
  - address: ip-10-0-131-183.ec2.internal
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Node
    name: ip-10-0-131-183.ec2.internal
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: awsproviderconfig.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: AWSMachineProviderStatus
```

ii. Changez le nom du champ **metadata.name**.
Il est recommandé de conserver le même nom de base que l'ancienne machine et de remplacer le numéro de fin par le prochain numéro disponible. Dans cet exemple, **clustername-8qw5l-master-0** est remplacé par **clustername-8qw5l-master-3**:

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. Supprimer le champ **spec.providerID**:

```
providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

iv. Supprimer les champs **metadata.annotations** et **metadata.generation**:

```
annotations:
  machine.openshift.io/instance-state: running
...
generation: 2
```

v. Supprimer les champs **metadata.resourceVersion** et **metadata.uid**:

```
resourceVersion: "13291"
uid: a282eb70-40a2-4e89-8009-d05dd420d31a
```

d. Supprimer la machine de l'hôte du plan de contrôle perdu :

```
oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 Indiquez le nom de la machine du plan de contrôle pour l'hôte du plan de contrôle perdu.

e. Vérifiez que la machine a été supprimée :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie :

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b   3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running   m4.xlarge   us-east-1   us-east-1c   3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a   running
```

f. Créez une machine en utilisant le fichier **new-master-machine.yaml**:

```
$ oc apply -f new-master-machine.yaml
```

g. Vérifiez que la nouvelle machine a été créée :

```
$ oc get machines -n openshift-machine-api -o wide
```

Exemple de sortie :

```
NAME                                PHASE         TYPE        REGION      ZONE
AGE   NODE                          PROVIDERID                          STATE
clustername-8qw5l-master-1          Running       m4.xlarge   us-east-1   us-east-
1b    3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631
running
clustername-8qw5l-master-2          Running       m4.xlarge   us-east-1   us-east-
1c    3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba
running
clustername-8qw5l-master-3          Provisioning  m4.xlarge   us-east-1   us-east-
1a    85s     ip-10-0-173-171.ec2.internal   aws:///us-east-1a/i-015b0888fe17bc2c8
running (1)
clustername-8qw5l-worker-us-east-1a-wbtgd  Running     m4.large    us-east-1
us-east-1a    3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-
010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running     m4.large    us-east-1   us-
east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-
0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26  Running     m4.large    us-east-1
us-east-1c    3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-
06861c00007751b0a   running
```

(1) La nouvelle machine, **clustername-8qw5l-master-3**, est en cours de création et sera prête après le changement de phase de **Provisioning** à **Running**.

La création de la nouvelle machine peut prendre quelques minutes. L'opérateur du cluster etcd se synchronisera automatiquement lorsque la machine ou le nœud reviendra à un état sain.

h. Répétez ces étapes pour chaque hôte du plan de contrôle perdu qui n'est pas l'hôte de récupération.

17. Dans une fenêtre de terminal séparée, connectez-vous au cluster en tant qu'utilisateur ayant le rôle **cluster-admin** en entrant la commande suivante :

```
$ oc login -u <cluster_admin> (1)
```

(1) Pour **<cluster_admin>**, indiquez un nom d'utilisateur avec le rôle **cluster-admin**.

18. Forcer le redéploiement de etcd.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc patch etcd cluster -p='{"spec" : {\i1}"forceRedeploymentReason" : \N-"recovery-'\N"$(
date --rfc-3339=ns )'\N'\N"}" -type=merge} --type=merge (1)
```

(1) La valeur **forceRedeploymentReason** doit être unique, c'est pourquoi un horodatage est ajouté.

Lorsque l'opérateur du cluster etcd effectue un redéploiement, les nœuds existants sont démarrés avec de nouveaux pods, comme lors de la mise à l'échelle initiale.

19. Réactivez la garde du quorum en entrant la commande suivante :

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

20. Vous pouvez vérifier que la section **unsupportedConfigOverrides** est supprimée de l'objet en entrant cette commande :

```
$ oc get etcd/cluster -oyaml
```

21. Vérifier que tous les nœuds sont mis à jour avec la dernière révision.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

Examinez la condition d'état **NodeInstallerProgressing** pour etcd afin de vérifier que tous les nœuds sont à la dernière révision. La sortie indique **AllNodesAtLatestRevision** lorsque la mise à jour est réussie :

```
AllNodesAtLatestRevision
3 nodes are at revision 7 (1)
```

(1) Dans cet exemple, le dernier numéro de révision est **7**.

Si le résultat comprend plusieurs numéros de révision, tels que **2 nodes are at revision 6; 1 nodes are at revision 7**, cela signifie que la mise à jour est toujours en cours. Attendez quelques minutes et réessayez.

22. Une fois etcd redéployé, forcez de nouveaux déploiements pour le plan de contrôle. Le serveur API Kubernetes se réinstallera sur les autres nœuds car le kubelet est connecté aux serveurs API à l'aide d'un équilibreur de charge interne.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez les commandes suivantes.

    a. Forcer un nouveau déploiement pour le serveur API Kubernetes :

    ```
    $ oc patch kubeapiserver cluster -p='{"spec": {"forceRedeploymentReason": "recovery-
    '"$( date --rfc-3339=ns )"'"}}' --type=merge
    ```

    Vérifier que tous les nœuds sont mis à jour avec la dernière révision.

```
$ oc get kubeapiserver -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

Examinez l'état de **NodeInstallerProgressing** pour vérifier que tous les nœuds sont à la dernière révision. La sortie indique **AllNodesAtLatestRevision** lorsque la mise à jour est réussie :

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 Dans cet exemple, le dernier numéro de révision est **7**.

Si le résultat comprend plusieurs numéros de révision, tels que **2 nodes are at revision 6; 1 nodes are at revision 7**, cela signifie que la mise à jour est toujours en cours. Attendez quelques minutes et réessayez.

b. Forcer un nouveau déploiement pour le gestionnaire de contrôleur Kubernetes :

```
$ oc patch kubecontrollermanager cluster -p='{"spec": {"forceRedeploymentReason":
"recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

Vérifier que tous les nœuds sont mis à jour avec la dernière révision.

```
$ oc get kubecontrollermanager -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

Examinez l'état de **NodeInstallerProgressing** pour vérifier que tous les nœuds sont à la dernière révision. La sortie indique **AllNodesAtLatestRevision** lorsque la mise à jour est réussie :

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 Dans cet exemple, le dernier numéro de révision est **7**.

Si le résultat comprend plusieurs numéros de révision, tels que **2 nodes are at revision 6; 1 nodes are at revision 7**, cela signifie que la mise à jour est toujours en cours. Attendez quelques minutes et réessayez.

c. Forcer un nouveau déploiement pour le planificateur Kubernetes :

```
$ oc patch kubescheduler cluster -p='{"spec": {"forceRedeploymentReason": "recovery-
'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

Vérifier que tous les nœuds sont mis à jour avec la dernière révision.

```
$ oc get kubescheduler -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

Examinez l'état de **NodeInstallerProgressing** pour vérifier que tous les nœuds sont à la dernière révision. La sortie indique **AllNodesAtLatestRevision** lorsque la mise à jour est réussie :

```
AllNodesAtLatestRevision
3 nodes are at revision 7 (1)
```

(1) Dans cet exemple, le dernier numéro de révision est **7**.

Si le résultat comprend plusieurs numéros de révision, tels que **2 nodes are at revision 6; 1 nodes are at revision 7**, cela signifie que la mise à jour est toujours en cours. Attendez quelques minutes et réessayez.

23. Vérifiez que tous les hôtes du plan de contrôle ont démarré et rejoint le cluster.
Dans un terminal ayant accès au cluster en tant qu'utilisateur **cluster-admin**, exécutez la commande suivante :

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

Exemple de sortie

```
etcd-ip-10-0-143-125.ec2.internal   2/2   Running   0   9h
etcd-ip-10-0-154-194.ec2.internal   2/2   Running   0   9h
etcd-ip-10-0-173-171.ec2.internal   2/2   Running   0   9h
```

Pour s'assurer que toutes les charges de travail reviennent à un fonctionnement normal à la suite d'une procédure de récupération, redémarrez chaque pod qui stocke les informations de l'API Kubernetes. Cela inclut les composants d'OpenShift Container Platform tels que les routeurs, les opérateurs et les composants tiers.

Notez que la restauration de tous les services peut prendre plusieurs minutes après l'exécution de cette procédure. Par exemple, l'authentification à l'aide de **oc login** peut ne pas fonctionner immédiatement jusqu'à ce que les pods du serveur OAuth soient redémarrés.

5.4.2.3. Ressources complémentaires

- Installation d'un cluster fourni par l'utilisateur sur du métal nu
- Création d'un hôte bastion pour accéder aux instances OpenShift Container Platform et aux nœuds du plan de contrôle avec SSH
- Remplacement d'un nœud de plan de contrôle nu

5.4.2.4. Problèmes et solutions de contournement pour la restauration d'un état de stockage persistant

Si votre cluster OpenShift Container Platform utilise un stockage persistant sous quelque forme que ce soit, un état du cluster est généralement stocké en dehors d'etcd. Il peut s'agir d'un cluster Elasticsearch fonctionnant dans un pod ou d'une base de données fonctionnant dans un objet **StatefulSet**. Lorsque vous restaurez à partir d'une sauvegarde etcd, l'état des charges de travail dans OpenShift Container Platform est également restauré. Cependant, si le snapshot etcd est ancien, l'état peut être invalide ou obsolète.

IMPORTANT

Le contenu des volumes persistants (PV) ne fait jamais partie de l'instantané etcd. Lorsque vous restaurez un cluster OpenShift Container Platform à partir d'un instantané etcd, les charges de travail non critiques peuvent avoir accès aux données critiques, et vice-versa.

Voici quelques exemples de scénarios qui produisent un état périmé :

- La base de données MySQL s'exécute dans un pod sauvegardé par un objet PV. La restauration d'OpenShift Container Platform à partir d'un snapshot etcd ne rétablit pas le volume sur le fournisseur de stockage et ne produit pas de pod MySQL en cours d'exécution, malgré les tentatives répétées de démarrage du pod. Vous devez restaurer manuellement ce pod en restaurant le volume sur le fournisseur de stockage, puis en modifiant le PV pour qu'il pointe vers le nouveau volume.

- Le pod P1 utilise le volume A, qui est attaché au nœud X. Si l'instantané etcd est pris alors qu'un autre pod utilise le même volume sur le nœud Y, alors lorsque la restauration etcd est effectuée, le pod P1 pourrait ne pas être en mesure de démarrer correctement en raison du fait que le volume est toujours attaché au nœud Y. OpenShift Container Platform n'est pas conscient de l'attachement et ne le détache pas automatiquement. Lorsque cela se produit, le volume doit être détaché manuellement du nœud Y afin que le volume puisse s'attacher au nœud X, puis le pod P1 peut démarrer.

- Les informations d'identification du fournisseur de cloud ou du fournisseur de stockage ont été mises à jour après que l'instantané etcd a été pris. Par conséquent, les pilotes ou opérateurs CSI qui dépendent de ces informations d'identification ne fonctionnent pas. Il se peut que vous deviez mettre à jour manuellement les informations d'identification requises par ces pilotes ou opérateurs.

- Un périphérique est supprimé ou renommé à partir des nœuds OpenShift Container Platform après que l'instantané etcd a été pris. L'opérateur de stockage local crée des liens symboliques pour chaque PV qu'il gère à partir des répertoires **/dev/disk/by-id** ou **/dev**. Cette situation peut amener les PV locaux à faire référence à des périphériques qui n'existent plus.
  Pour résoudre ce problème, l'administrateur doit

  1. Supprimer manuellement les PV dont les dispositifs ne sont pas valides.

  2. Supprimer les liens symboliques des nœuds respectifs.

  3. Supprimer les objets **LocalVolume** ou **LocalVolumeSet** (voir *Storage → Configuring persistent storage → Persistent storage using local volumes → Deleting the Local Storage Operator Resources*).

5.4.3. Récupération des certificats expirés du plan de contrôle

5.4.3.1. Récupération des certificats expirés du plan de contrôle

Le cluster peut récupérer automatiquement les certificats du plan de contrôle qui ont expiré.

Cependant, vous devez approuver manuellement les demandes de signature de certificat (CSR) en attente sur **node-bootstrapper** pour récupérer les certificats des kubelets. Pour les installations fournies par l'utilisateur, il se peut que vous deviez également approuver les CSR de service des kubelets en attente.

Procédez comme suit pour approuver les CSR en attente :

Procédure

1. Obtenir la liste des CSR actuels :

```
$ oc get csr
```

Exemple de sortie

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2s94x   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending (1)
csr-4bd6t   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending (2)
csr-4hl85   13m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending (3)
csr-zhhhp   3m8s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending (4)
...
```

(1) (2) Un CSR de service kubelet en attente (pour les installations fournies par l'utilisateur).

(3) (4) Un CSR **node-bootstrapper** en attente.

2. Examinez les détails d'un CSR pour vérifier qu'il est valide :

```
oc describe csr <csr_name> (1)
```

(1) **<csr_name>** est le nom d'un CSR figurant dans la liste des CSR actuels.

3. Approuver chaque CSR **node-bootstrapper** valide :

```
$ oc adm certificate approve <csr_name>
```

4. Pour les installations fournies par l'utilisateur, approuver chaque kubelet valide servant de CSR :

```
$ oc adm certificate approve <csr_name>
```

5.4.4. Reprise après sinistre pour un cluster hébergé dans une région AWS

Si vous avez besoin d'une reprise après sinistre (DR) pour un cluster hébergé, vous pouvez récupérer un cluster hébergé dans la même région au sein d'AWS. Par exemple, vous avez besoin d'une reprise après sinistre lorsque la mise à niveau d'un cluster de gestion échoue et que le cluster hébergé est en lecture seule.

IMPORTANT

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

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

Le processus de DR comporte trois étapes principales :

1. Sauvegarde du cluster hébergé sur le cluster de gestion source
2. Restauration du cluster hébergé sur un cluster de gestion de destination
3. Suppression du cluster hébergé du cluster de gestion source

Vos charges de travail restent en cours d'exécution pendant le processus. L'API du cluster peut être indisponible pendant un certain temps, mais cela n'affectera pas les services exécutés sur les nœuds de travail.

IMPORTANT

Le cluster de gestion source et le cluster de gestion de destination doivent avoir les drapeaux **--external-dns** pour gérer l'URL du serveur API, comme indiqué dans cet exemple :

Exemple : Drapeaux DNS externes

```
--external-dns-provider=aws \
--external-dns-credentials=<AWS Credentials location> \
--external-dns-domain-filter=<DNS Base Domain>
```

Ainsi, l'URL du serveur se termine par **https://api-sample-hosted.sample-hosted.aws.openshift.com**.

Si vous n'incluez pas les drapeaux **--external-dns** pour maintenir l'URL du serveur API, le cluster hébergé ne peut pas être migré.

5.4.4.1. Exemple d'environnement et de contexte

Considérons un scénario dans lequel vous avez trois clusters à restaurer. Deux sont des clusters de gestion et un est un cluster hébergé. Vous pouvez restaurer soit le plan de contrôle uniquement, soit le plan de contrôle et les nœuds. Avant de commencer, vous avez besoin des informations suivantes :

- Espace de noms Source MGMT : L'espace de noms de la gestion de la source
- Source MGMT ClusterName : Le nom du cluster de gestion source
- Source MGMT Kubeconfig : Le fichier de gestion des sources **kubeconfig**

- Destination MGMT Kubeconfig : Le fichier de gestion de destination **kubeconfig**
- HC Kubeconfig : Le fichier du cluster hébergé **kubeconfig**
- Fichier de clé SSH : La clé publique SSH
- Secret d'extraction : le fichier secret d'extraction pour accéder aux images de la version
- Références AWS
- Région AWS
- Domaine de base : Le domaine de base DNS à utiliser comme domaine DNS externe
- Nom du seau S3 : Le seau dans la région AWS où vous prévoyez de télécharger la sauvegarde etcd

Ces informations sont présentées dans l'exemple suivant de variables d'environnement.

Exemple de variables d'environnement

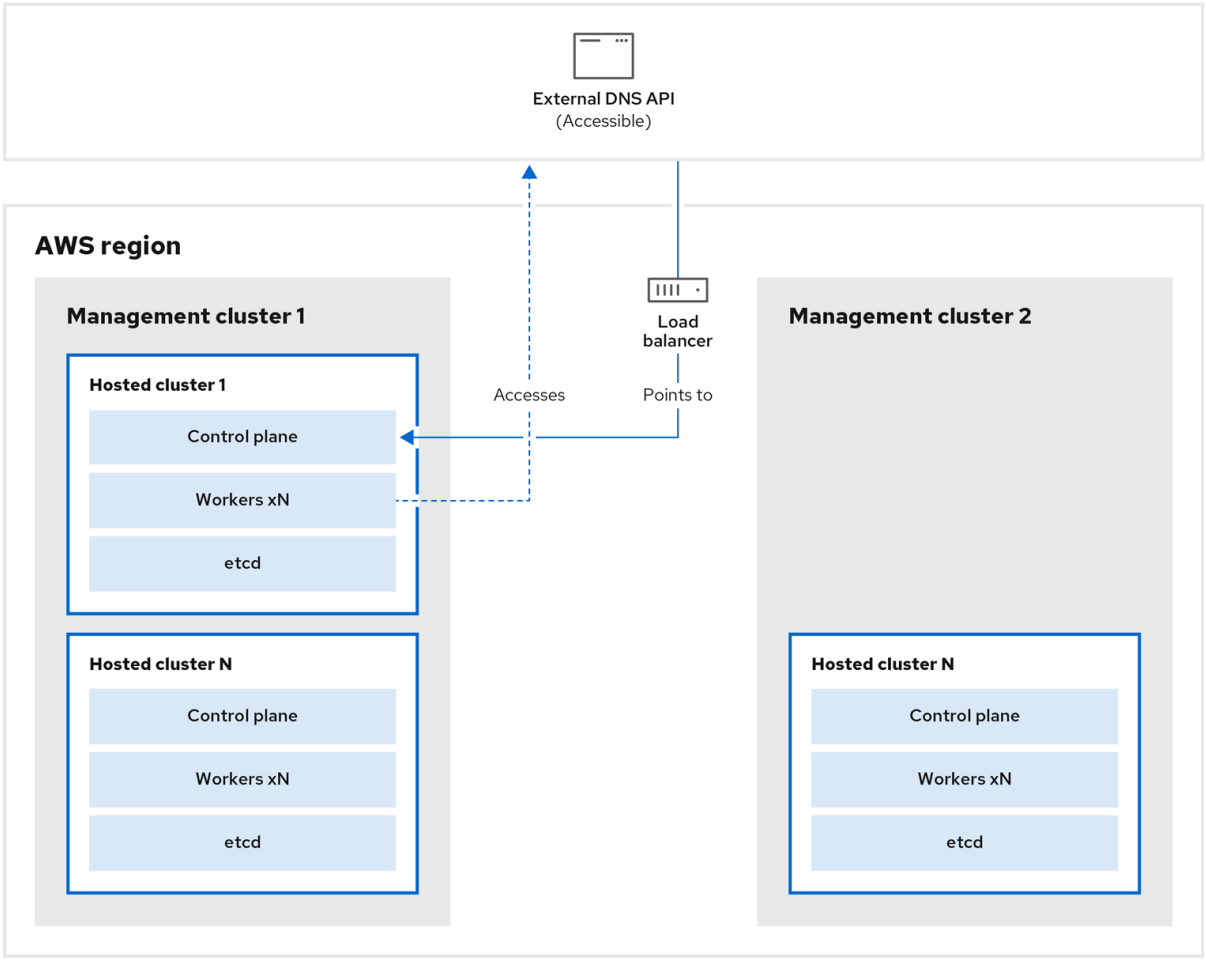
```
SSH_KEY_FILE=${HOME}/.ssh/id_rsa.pub
BASE_PATH=${HOME}/hypershift
BASE_DOMAIN="aws.sample.com"
PULL_SECRET_FILE="${HOME}/pull_secret.json"
AWS_CREDS="${HOME}/.aws/credentials"
AWS_ZONE_ID="Z02718293M33QHDEQBROL"

CONTROL_PLANE_AVAILABILITY_POLICY=SingleReplica
HYPERSHIFT_PATH=${BASE_PATH}/src/hypershift
HYPERSHIFT_CLI=${HYPERSHIFT_PATH}/bin/hypershift
HYPERSHIFT_IMAGE=${HYPERSHIFT_IMAGE:-"quay.io/${USER}/hypershift:latest"}
NODE_POOL_REPLICAS=${NODE_POOL_REPLICAS:-2}

# MGMT Context
MGMT_REGION=us-west-1
MGMT_CLUSTER_NAME="${USER}-dev"
MGMT_CLUSTER_NS=${USER}
MGMT_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${MGMT_CLUSTER_NS}-
${MGMT_CLUSTER_NAME}"
MGMT_KUBECONFIG="${MGMT_CLUSTER_DIR}/kubeconfig"

# MGMT2 Context
MGMT2_CLUSTER_NAME="${USER}-dest"
MGMT2_CLUSTER_NS=${USER}
MGMT2_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${MGMT2_CLUSTER_NS}-
${MGMT2_CLUSTER_NAME}"
MGMT2_KUBECONFIG="${MGMT2_CLUSTER_DIR}/kubeconfig"

# Hosted Cluster Context
HC_CLUSTER_NS=clusters
HC_REGION=us-west-1
HC_CLUSTER_NAME="${USER}-hosted"
HC_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}"
HC_KUBECONFIG="${HC_CLUSTER_DIR}/kubeconfig"
```

```
BACKUP_DIR=${HC_CLUSTER_DIR}/backup

BUCKET_NAME="${USER}-hosted-${MGMT_REGION}"

# DNS
AWS_ZONE_ID="Z07342811SH9AA102K1AC"
EXTERNAL_DNS_DOMAIN="hc.jpdv.aws.kerbeross.com"
```

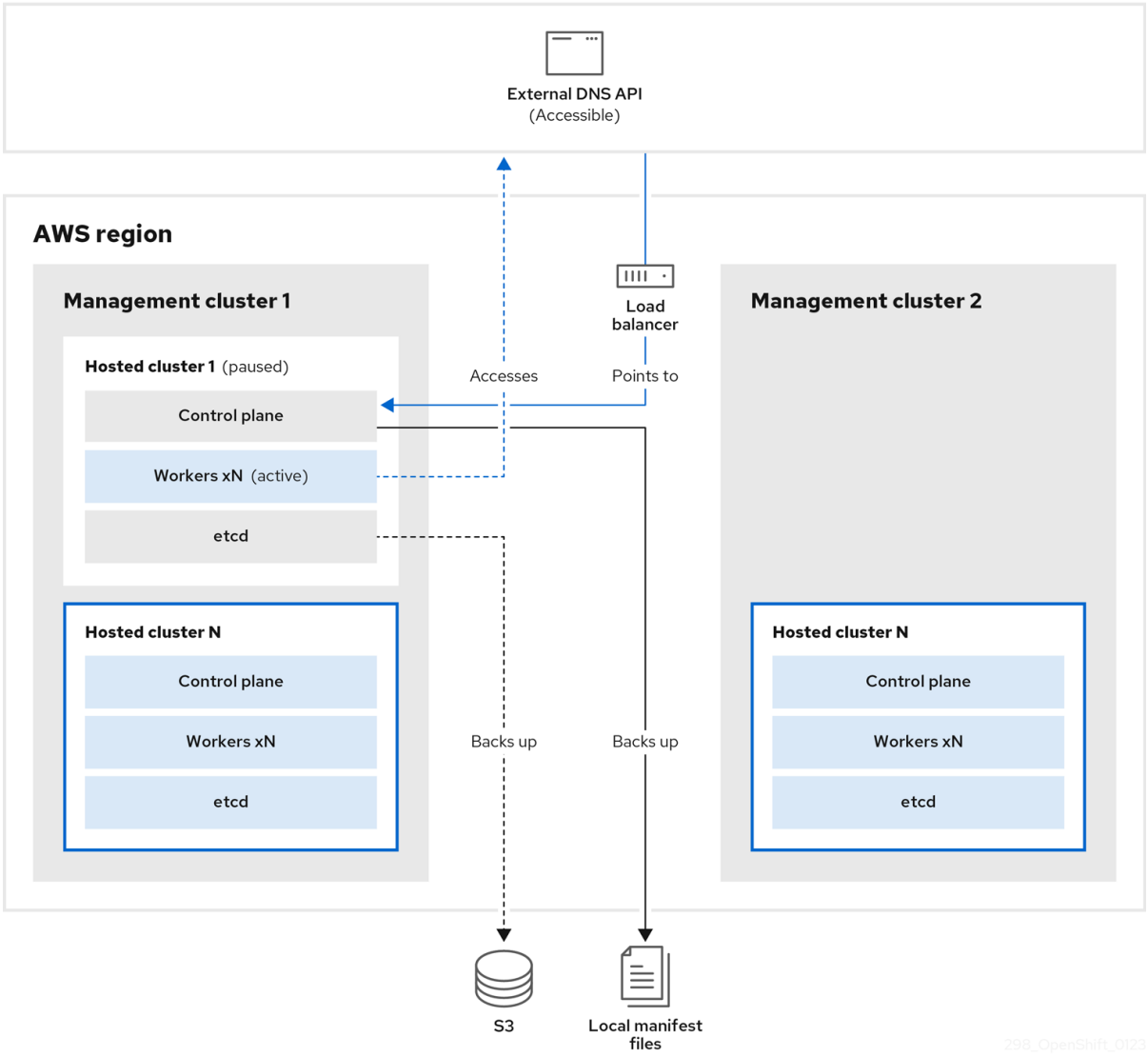
5.4.4.2. Vue d'ensemble du processus de sauvegarde et de restauration

Le processus de sauvegarde et de restauration fonctionne comme suit :

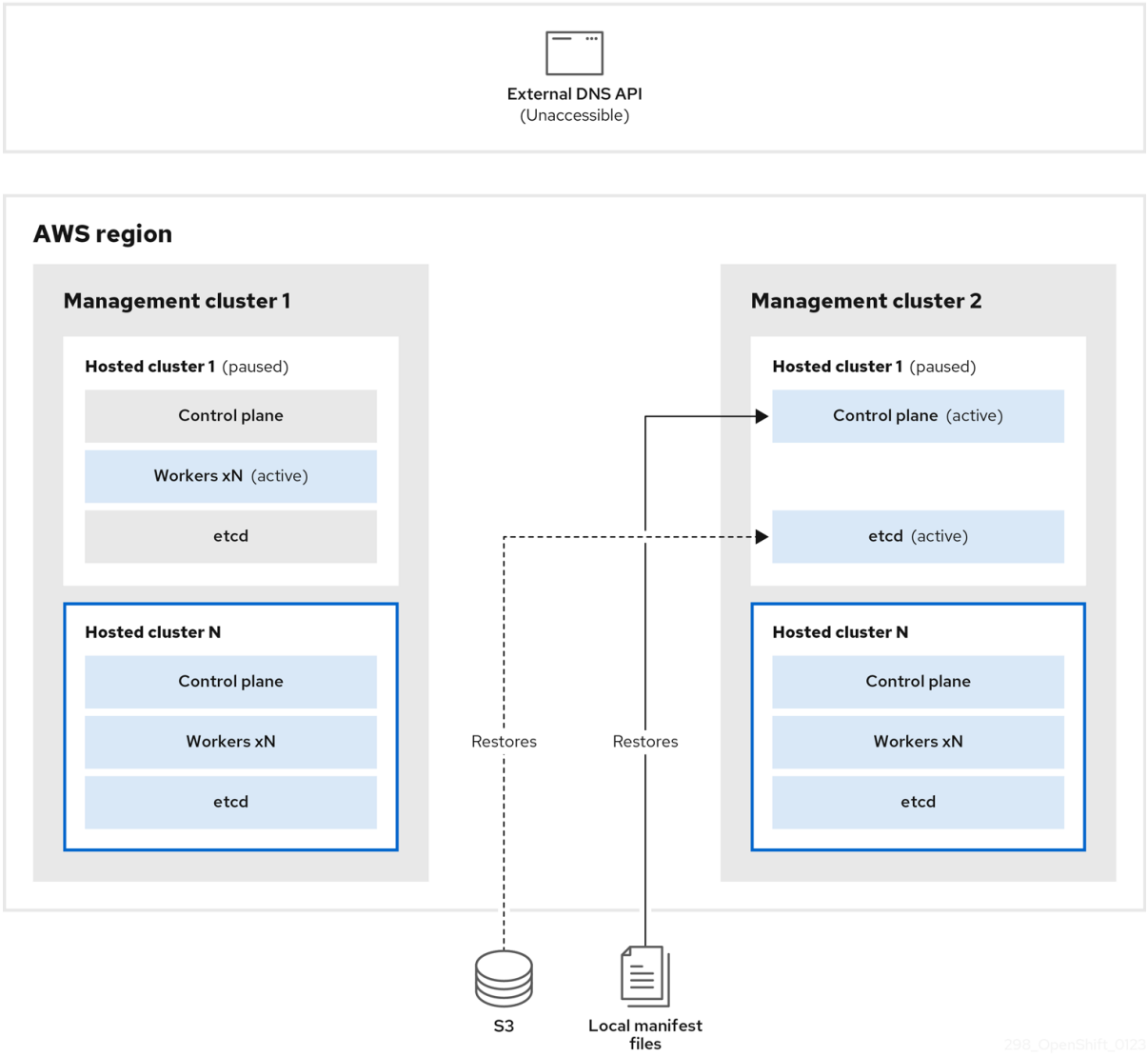
1. Sur le cluster de gestion 1, que vous pouvez considérer comme le cluster de gestion source, le plan de contrôle et les travailleurs interagissent en utilisant l'API DNS externe. L'API DNS externe est accessible et un équilibreur de charge se trouve entre les clusters de gestion.



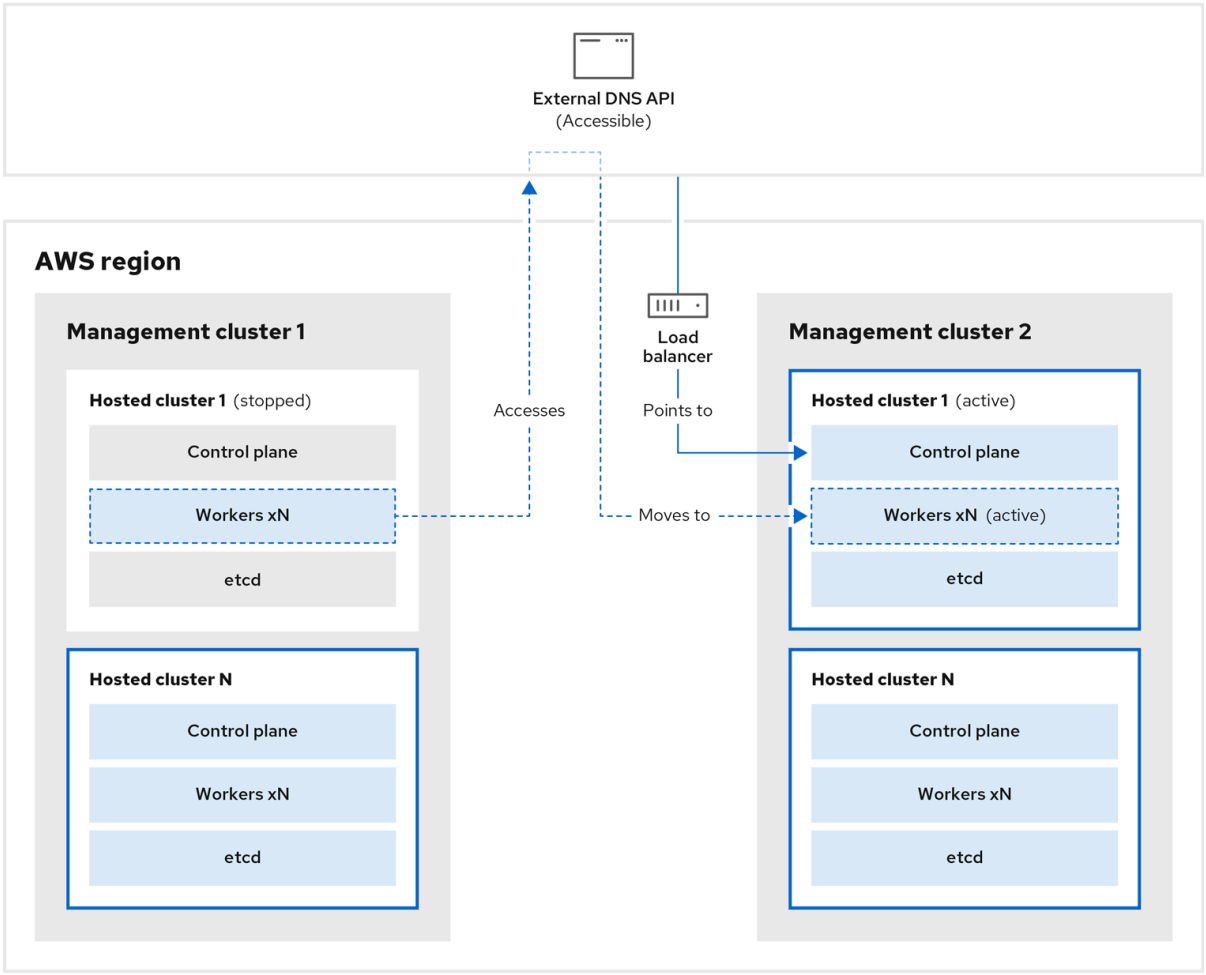
2. Vous prenez un instantané du cluster hébergé, qui comprend etcd, le plan de contrôle et les nœuds de travail. Au cours de ce processus, les nœuds de travail continuent d'essayer d'accéder à l'API DNS externe même si elle n'est pas accessible, les charges de travail sont en cours d'exécution, le plan de contrôle est sauvegardé dans un fichier manifeste local et etcd est sauvegardé dans un seau S3. Le plan de données est actif et le plan de contrôle est en pause.



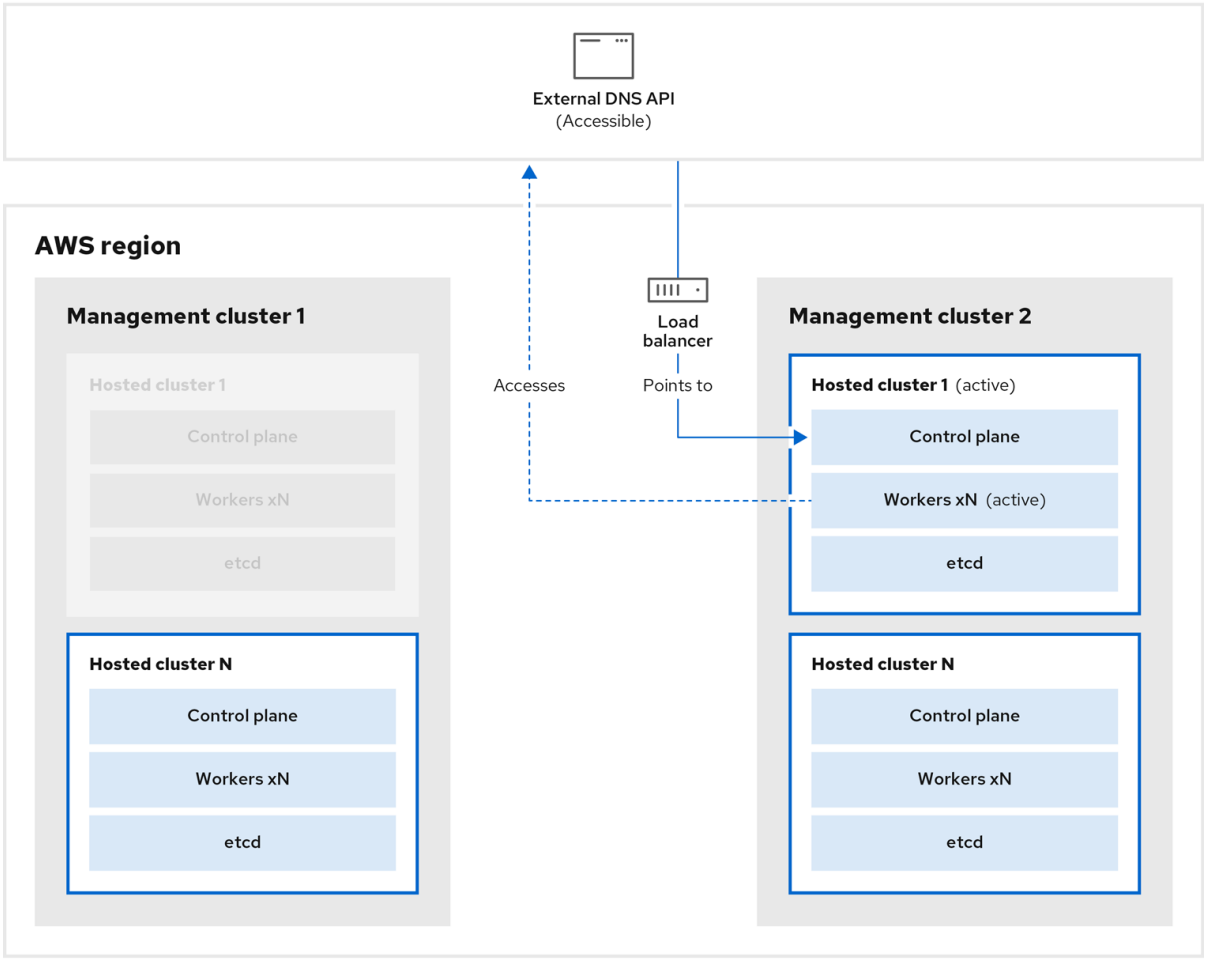
3. Sur le cluster de gestion 2, que vous pouvez considérer comme le cluster de gestion de destination, vous restaurez etcd à partir du seau S3 et restaurez le plan de contrôle à partir du fichier manifeste local. Au cours de ce processus, l'API DNS externe est arrêtée, l'API du cluster hébergé devient inaccessible et tous les travailleurs qui utilisent l'API sont incapables de mettre à jour leurs fichiers manifestes, mais les charges de travail sont toujours en cours d'exécution.



4. L'API DNS externe est à nouveau accessible et les nœuds de travail l'utilisent pour passer au cluster de gestion 2. L'API DNS externe peut accéder à l'équilibreur de charge qui pointe vers le plan de contrôle.



5. Sur le cluster de gestion 2, le plan de contrôle et les nœuds de travail interagissent en utilisant l'API DNS externe. Les ressources sont supprimées du cluster de gestion 1, à l'exception de la sauvegarde S3 de etcd. Si vous essayez de configurer à nouveau le cluster hébergé sur le cluster de gestion 1, cela ne fonctionnera pas.



Vous pouvez sauvegarder et restaurer manuellement votre cluster hébergé ou exécuter un script pour terminer le processus. Pour plus d'informations sur le script, voir "Exécution d'un script pour sauvegarder et restaurer un cluster hébergé".

5.4.4.3. Sauvegarde d'un cluster hébergé

Pour récupérer votre cluster hébergé dans votre cluster de gestion cible, vous devez d'abord sauvegarder toutes les données pertinentes.

Procédure

1. Créez un fichier configmap pour déclarer le cluster de gestion des sources en entrant cette commande :

```
$ oc create configmap mgmt-parent-cluster -n default --from-
literal=from=${MGMT_CLUSTER_NAME}
```

2. Arrêtez la réconciliation dans le cluster hébergé et dans les pools de nœuds en entrant ces commandes :

```
PAUSED_UNTIL="true"
oc patch -n ${HC_CLUSTER_NS} hostedclusters/${HC_CLUSTER_NAME} -p '{"spec":
{"pausedUntil":"${PAUSED_UNTIL}"}}' --type=merge
oc scale deployment -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --replicas=0 kube-
apiserver openshift-apiserver openshift-oauth-apiserver control-plane-operator
```

```
PAUSED_UNTIL="true"
oc patch -n ${HC_CLUSTER_NS} hostedclusters/${HC_CLUSTER_NAME} -p '{"spec":
{"pausedUntil":"${PAUSED_UNTIL}"}}' --type=merge
oc patch -n ${HC_CLUSTER_NS} nodepools/${NODEPOOLS} -p '{"spec":
{"pausedUntil":"${PAUSED_UNTIL}"}}' --type=merge
oc scale deployment -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --replicas=0 kube-
apiserver openshift-apiserver openshift-oauth-apiserver control-plane-operator
```

3. Sauvegardez etcd et téléchargez les données vers un panier S3 en exécutant ce script bash :

ASTUCE

Enveloppez ce script dans une fonction et appelez-le à partir de la fonction principale.

```
# ETCD Backup
ETCD_PODS="etcd-0"
if [ "${CONTROL_PLANE_AVAILABILITY_POLICY}" = "HighlyAvailable" ]; then
  ETCD_PODS="etcd-0 etcd-1 etcd-2"
fi

for POD in ${ETCD_PODS}; do
  # Create an etcd snapshot
  oc exec -it ${POD} -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -- env
ETCDCTL_API=3 /usr/bin/etcdctl --cacert /etc/etcd/tls/client/etcd-client-ca.crt --cert
/etc/etcd/tls/client/etcd-client.crt --key /etc/etcd/tls/client/etcd-client.key --
endpoints=localhost:2379 snapshot save /var/lib/data/snapshot.db
  oc exec -it ${POD} -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -- env
ETCDCTL_API=3 /usr/bin/etcdctl -w table snapshot status /var/lib/data/snapshot.db

  FILEPATH="/${BUCKET_NAME}/${HC_CLUSTER_NAME}-${POD}-snapshot.db"
  CONTENT_TYPE="application/x-compressed-tar"
  DATE_VALUE=`date -R`
  SIGNATURE_STRING="PUT\n\n${CONTENT_TYPE}\n${DATE_VALUE}\n${FILEPATH}"

  set +x
  ACCESS_KEY=$(grep aws_access_key_id ${AWS_CREDS} | head -n1 | cut -d= -f2 | sed
"s/ //g")
  SECRET_KEY=$(grep aws_secret_access_key ${AWS_CREDS} | head -n1 | cut -d= -f2 |
sed "s/ //g")
  SIGNATURE_HASH=$(echo -en ${SIGNATURE_STRING} | openssl sha1 -hmac
"${SECRET_KEY}" -binary | base64)
  set -x

  # FIXME: this is pushing to the OIDC bucket
  oc exec -it etcd-0 -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -- curl -X PUT -T
"/var/lib/data/snapshot.db" \
    -H "Host: ${BUCKET_NAME}.s3.amazonaws.com" \
    -H "Date: ${DATE_VALUE}" \
    -H "Content-Type: ${CONTENT_TYPE}" \
    -H "Authorization: AWS ${ACCESS_KEY}:${SIGNATURE_HASH}" \
    https://${BUCKET_NAME}.s3.amazonaws.com/${HC_CLUSTER_NAME}-${POD}-
snapshot.db
done
```

Pour plus d'informations sur la sauvegarde d'etcd, voir "Sauvegarde et restauration d'etcd sur un cluster hébergé".

4. Sauvegardez les objets Kubernetes et OpenShift Container Platform en entrant les commandes suivantes. Vous devez sauvegarder les objets suivants :

   - **HostedCluster** et **NodePool** objets de l'espace de noms HostedCluster
   - **HostedCluster** secrets de l'espace de noms HostedCluster
   - **HostedControlPlane** de l'espace de noms du plan de contrôle hébergé
   - **Cluster** de l'espace de noms du plan de contrôle hébergé
   - **AWSCluster**, **AWSMachineTemplate**, et **AWSMachine** de l'espace de noms du plan de contrôle hébergé
   - **MachineDeployments**, **MachineSets**, et **Machines** de l'espace de noms du plan de contrôle hébergé
   - **ControlPlane** secrets de l'espace de noms du plan de contrôle hébergé

```
mkdir -p ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
chmod 700 ${BACKUP_DIR}/namespaces/

# HostedCluster
echo "Backing Up HostedCluster Objects:"
oc get hc ${HC_CLUSTER_NAME} -n ${HC_CLUSTER_NS} -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/hc-${HC_CLUSTER_NAME}.yaml
echo "--> HostedCluster"
sed -i '' -e '/^status:$/,$d' ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/hc-
${HC_CLUSTER_NAME}.yaml

# NodePool
oc get np ${NODEPOOLS} -n ${HC_CLUSTER_NS} -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/np-${NODEPOOLS}.yaml
echo "--> NodePool"
sed -i '' -e '/^status:$/,$ d' ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/np-
${NODEPOOLS}.yaml

# Secrets in the HC Namespace
echo "--> HostedCluster Secrets:"
for s in $(oc get secret -n ${HC_CLUSTER_NS} | grep "^${HC_CLUSTER_NAME}" |
awk '{print $1}'); do
    oc get secret -n ${HC_CLUSTER_NS} $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/secret-${s}.yaml
done

# Secrets in the HC Control Plane Namespace
echo "--> HostedCluster ControlPlane Secrets:"
for s in $(oc get secret -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} | egrep -v
"docker|service-account-token|oauth-openshift|NAME|token-${HC_CLUSTER_NAME}" |
awk '{print $1}'); do
    oc get secret -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}/secret-
${s}.yaml
```

```
done

# Hosted Control Plane
echo "--> HostedControlPlane:"
oc get hcp ${HC_CLUSTER_NAME} -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
-o yaml > ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/hcp-${HC_CLUSTER_NAME}.yaml

# Cluster
echo "--> Cluster:"
CL_NAME=$(oc get hcp ${HC_CLUSTER_NAME} -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} -o jsonpath={.metadata.labels.\*} | grep
${HC_CLUSTER_NAME})
oc get cluster ${CL_NAME} -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -o yaml
> ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}/cl-
${HC_CLUSTER_NAME}.yaml

# AWS Cluster
echo "--> AWS Cluster:"
oc get awscluster ${HC_CLUSTER_NAME} -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}/awscl-
${HC_CLUSTER_NAME}.yaml

# AWS MachineTemplate
echo "--> AWS Machine Template:"
oc get awsmachinetemplate ${NODEPOOLS} -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}/awsmt-
${HC_CLUSTER_NAME}.yaml

# AWS Machines
echo "--> AWS Machine:"
CL_NAME=$(oc get hcp ${HC_CLUSTER_NAME} -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} -o jsonpath={.metadata.labels.\*} | grep
${HC_CLUSTER_NAME})
for s in $(oc get awsmachines -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --no-
headers | grep ${CL_NAME} | cut -f1 -d\ ); do
   oc get -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} awsmachines $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}/awsm-
${s}.yaml
done

# MachineDeployments
echo "--> HostedCluster MachineDeployments:"
for s in $(oc get machinedeployment -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
-o name); do
   mdp_name=$(echo ${s} | cut -f 2 -d /)
   oc get -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machinedeployment-${mdp_name}.yaml
done

# MachineSets
echo "--> HostedCluster MachineSets:"
for s in $(oc get machineset -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -o
```

```
name); do
    ms_name=$(echo ${s} | cut -f 2 -d /)
    oc get -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machineset-${ms_name}.yaml
done

# Machines
echo "--> HostedCluster Machine:"
for s in $(oc get machine -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -o name);
do
    m_name=$(echo ${s} | cut -f 2 -d /)
    oc get -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} $s -o yaml >
${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machine-${m_name}.yaml
done
```

5. Nettoyez les itinéraires **ControlPlane** en entrant cette commande :

```
$ oc delete routes -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --all
```

En entrant cette commande, vous permettez à l'opérateur ExternalDNS de supprimer les entrées Route53.

6. Vérifiez que les entrées de Route53 sont propres en exécutant ce script :

```
function clean_routes() {

    if [[ -z "${1}" ]];then
        echo "Give me the NS where to clean the routes"
        exit 1
    fi

    # Constants
    if [[ -z "${2}" ]];then
        echo "Give me the Route53 zone ID"
        exit 1
    fi

    ZONE_ID=${2}
    ROUTES=10
    timeout=40
    count=0

    # This allows us to remove the ownership in the AWS for the API route
    oc delete route -n ${1} --all

    while [ ${ROUTES} -gt 2 ]
    do
        echo "Waiting for ExternalDNS Operator to clean the DNS Records in AWS Route53
where the zone id is: ${ZONE_ID}..."
        echo "Try: (${count}/${timeout})"
        sleep 10
        if [[ $count -eq timeout ]];then
            echo "Timeout waiting for cleaning the Route53 DNS records"
            exit 1
```

```
        fi
        count=$((count+1))
        ROUTES=$(aws route53 list-resource-record-sets --hosted-zone-id ${ZONE_ID} --max-
items 10000 --output json | grep -c ${EXTERNAL_DNS_DOMAIN})
    done
}

# SAMPLE: clean_routes "<HC ControlPlane Namespace>" "<AWS_ZONE_ID>"
clean_routes "${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}" "${AWS_ZONE_ID}"
```

Vérification

Vérifiez tous les objets d'OpenShift Container Platform et le seau S3 pour vous assurer que tout se passe comme prévu.

Prochaines étapes

Restaurez votre cluster hébergé.

5.4.4.4. Restauration d'un cluster hébergé

Rassemblez tous les objets que vous avez sauvegardés et restaurez-les dans votre cluster de gestion de destination.

Conditions préalables

Vous avez sauvegardé les données de votre cluster de gestion des sources.

ASTUCE

Assurez-vous que le fichier **kubeconfig** du cluster de gestion de destination est placé tel qu'il est défini dans la variable **KUBECONFIG** ou, si vous utilisez le script, dans la variable **MGMT2_KUBECONFIG**. Utilisez **export KUBECONFIG=<Kubeconfig FilePath>** ou, si vous utilisez le script, **export KUBECONFIG=${MGMT2_KUBECONFIG}**.

Procédure

1. Vérifiez que le nouveau cluster de gestion ne contient aucun espace de noms du cluster que vous restaurez en entrant ces commandes :

```
# Just in case
export KUBECONFIG=${MGMT2_KUBECONFIG}
BACKUP_DIR=${HC_CLUSTER_DIR}/backup

# Namespace deletion in the destination Management cluster
$ oc delete ns ${HC_CLUSTER_NS} || true
$ oc delete ns ${HC_CLUSTER_NS}-{HC_CLUSTER_NAME} || true
```

2. Recréez les espaces de noms supprimés en entrant ces commandes :

```
# Namespace creation
$ oc new-project ${HC_CLUSTER_NS}
$ oc new-project ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
```

3. Rétablissez les secrets dans l'espace de noms HC en entrant cette commande :

```
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/secret-*
```

4. Restaurez les objets dans l'espace de noms du plan de contrôle **HostedCluster** en entrant ces commandes :

```
# Secrets
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/secret-*

# Cluster
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/hcp-*
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/cl-*
```

5. Si vous récupérez les nœuds et le pool de nœuds pour réutiliser les instances AWS, restaurez les objets dans l'espace de noms du plan de contrôle HC en entrant ces commandes :

```
# AWS
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/awscl-*
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/awsmt-*
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/awsm-*

# Machines
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machinedeployment-*
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machineset-*
$ oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}/machine-*
```

6. Restaurez les données etcd et le cluster hébergé en exécutant ce script bash :

```
ETCD_PODS="etcd-0"
if [ "${CONTROL_PLANE_AVAILABILITY_POLICY}" = "HighlyAvailable" ]; then
  ETCD_PODS="etcd-0 etcd-1 etcd-2"
fi

HC_RESTORE_FILE=${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/hc-
${HC_CLUSTER_NAME}-restore.yaml
HC_BACKUP_FILE=${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/hc-
${HC_CLUSTER_NAME}.yaml
HC_NEW_FILE=${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/hc-
${HC_CLUSTER_NAME}-new.yaml
cat ${HC_BACKUP_FILE} > ${HC_NEW_FILE}
cat > ${HC_RESTORE_FILE} <<EOF
    restoreSnapshotURL:
EOF

for POD in ${ETCD_PODS}; do
  # Create a pre-signed URL for the etcd snapshot
  ETCD_SNAPSHOT="s3://${BUCKET_NAME}/${HC_CLUSTER_NAME}-${POD}-
```

```
snapshot.db"
  ETCD_SNAPSHOT_URL=$(AWS_DEFAULT_REGION=${MGMT2_REGION} aws s3
presign ${ETCD_SNAPSHOT})

  # FIXME no CLI support for restoreSnapshotURL yet
  cat >> ${HC_RESTORE_FILE} <<EOF
    - "${ETCD_SNAPSHOT_URL}"
EOF
done

cat ${HC_RESTORE_FILE}

if ! grep ${HC_CLUSTER_NAME}-snapshot.db ${HC_NEW_FILE}; then
  sed -i '' -e "/type: PersistentVolume/r ${HC_RESTORE_FILE}" ${HC_NEW_FILE}
  sed -i '' -e '/pausedUntil:/d' ${HC_NEW_FILE}
fi

HC=$(oc get hc -n ${HC_CLUSTER_NS} ${HC_CLUSTER_NAME} -o name || true)
if [[ ${HC} == "" ]];then
    echo "Deploying HC Cluster: ${HC_CLUSTER_NAME} in ${HC_CLUSTER_NS}
namespace"
    oc apply -f ${HC_NEW_FILE}
else
    echo "HC Cluster ${HC_CLUSTER_NAME} already exists, avoiding step"
fi
```

7. Si vous récupérez les nœuds et le pool de nœuds pour réutiliser les instances AWS, restaurez le pool de nœuds en entrant cette commande :

```
oc apply -f ${BACKUP_DIR}/namespaces/${HC_CLUSTER_NS}/np-*
```

Vérification

- Pour vérifier que les nœuds sont entièrement restaurés, utilisez cette fonction :

```
timeout=40
count=0
NODE_STATUS=$(oc get nodes --kubeconfig=${HC_KUBECONFIG} | grep -v NotReady |
grep -c "worker") || NODE_STATUS=0

while [ ${NODE_POOL_REPLICAS} != ${NODE_STATUS} ]
do
    echo "Waiting for Nodes to be Ready in the destination MGMT Cluster:
${MGMT2_CLUSTER_NAME}"
    echo "Try: (${count}/${timeout})"
    sleep 30
    if [[ $count -eq timeout ]];then
        echo "Timeout waiting for Nodes in the destination MGMT Cluster"
        exit 1
    fi
    count=$((count+1))
    NODE_STATUS=$(oc get nodes --kubeconfig=${HC_KUBECONFIG} | grep -v NotReady |
grep -c "worker") || NODE_STATUS=0
done
```

Prochaines étapes

Arrêtez et supprimez votre cluster.

5.4.4.5. Suppression d'un cluster hébergé de votre cluster de gestion des sources

Après avoir sauvegardé votre cluster hébergé et l'avoir restauré sur votre cluster de gestion de destination, vous arrêtez et supprimez le cluster hébergé sur votre cluster de gestion source.

Conditions préalables

Vous avez sauvegardé vos données et les avez restaurées dans votre cluster de gestion des sources.

ASTUCE

Assurez-vous que le fichier **kubeconfig** du cluster de gestion de destination est placé tel qu'il est défini dans la variable **KUBECONFIG** ou, si vous utilisez le script, dans la variable **MGMT_KUBECONFIG**. Utilisez **export KUBECONFIG=<Kubeconfig FilePath>** ou, si vous utilisez le script, **export KUBECONFIG=${MGMT_KUBECONFIG}**.

Procédure

1. Mettez à l'échelle les objets **deployment** et **statefulset** en entrant ces commandes :

```
# Just in case
export KUBECONFIG=${MGMT_KUBECONFIG}

# Scale down deployments
oc scale deployment -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --replicas=0 --all
oc scale statefulset.apps -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --replicas=0 --
all
sleep 15
```

2. Supprimez les objets **NodePool** en entrant les commandes suivantes :

```
NODEPOOLS=$(oc get nodepools -n ${HC_CLUSTER_NS} -o=jsonpath='{.items[?
(@.spec.clusterName=="'${HC_CLUSTER_NAME}'")].metadata.name}')
if [[ ! -z "${NODEPOOLS}" ]];then
    oc patch -n "${HC_CLUSTER_NS}" nodepool ${NODEPOOLS} --type=json --patch='[ {
"op":"remove", "path": "/metadata/finalizers" }]'
    oc delete np -n ${HC_CLUSTER_NS} ${NODEPOOLS}
fi
```

3. Supprimez les objets **machine** et **machineset** en entrant ces commandes :

```
# Machines
for m in $(oc get machines -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -o name); do
    oc patch -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} ${m} --type=json --patch='[ {
"op":"remove", "path": "/metadata/finalizers" }]' || true
    oc delete -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} ${m} || true
done

oc delete machineset -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --all || true
```

4. Supprimez l'objet cluster en entrant ces commandes :

```
# Cluster
C_NAME=$(oc get cluster -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} -o name)
oc patch -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} ${C_NAME} --type=json --
patch='[ { "op":"remove", "path": "/metadata/finalizers" }]'
oc delete cluster.cluster.x-k8s.io -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} --all
```

5. Supprimez les machines AWS (objets Kubernetes) en entrant ces commandes. Ne vous préoccupez pas de la suppression des machines AWS réelles. Les instances cloud ne seront pas affectées.

```
# AWS Machines
for m in $(oc get awsmachine.infrastructure.cluster.x-k8s.io -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} -o name)
do
    oc patch -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} ${m} --type=json --patch='[ {
"op":"remove", "path": "/metadata/finalizers" }]' || true
    oc delete -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} ${m} || true
done
```

6. Supprimez les objets de l'espace de noms **HostedControlPlane** et **ControlPlane** HC en entrant ces commandes :

```
# Delete HCP and ControlPlane HC NS
oc patch -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
hostedcontrolplane.hypershift.openshift.io ${HC_CLUSTER_NAME} --type=json --patch='[ {
"op":"remove", "path": "/metadata/finalizers" }]'
oc delete hostedcontrolplane.hypershift.openshift.io -n ${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME} --all
oc delete ns ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME} || true
```

7. Supprimez les objets de l'espace de noms **HostedCluster** et HC en entrant ces commandes :

```
# Delete HC and HC Namespace
oc -n ${HC_CLUSTER_NS} patch hostedclusters ${HC_CLUSTER_NAME} -p '{"metadata":
{"finalizers":null}}' --type merge || true
oc delete hc -n ${HC_CLUSTER_NS} ${HC_CLUSTER_NAME}  || true
oc delete ns ${HC_CLUSTER_NS} || true
```

Vérification

- Pour vérifier que tout fonctionne, entrez les commandes suivantes :

```
# Validations
export KUBECONFIG=${MGMT2_KUBECONFIG}

oc get hc -n ${HC_CLUSTER_NS}
oc get np -n ${HC_CLUSTER_NS}
oc get pod -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}
oc get machines -n ${HC_CLUSTER_NS}-${HC_CLUSTER_NAME}

# Inside the HostedCluster
export KUBECONFIG=${HC_KUBECONFIG}
oc get clusterversion
oc get nodes
```

Prochaines étapes

Supprimez les pods OVN dans le cluster hébergé afin de pouvoir vous connecter au nouveau plan de contrôle OVN qui s'exécute dans le nouveau cluster de gestion :

1. Chargez la variable d'environnement **KUBECONFIG** avec le chemin kubeconfig du cluster hébergé.

2. Entrez cette commande :

```
$ oc delete pod -n openshift-ovn-kubernetes --all
```

5.4.4.6. Exécution d'un script pour sauvegarder et restaurer un cluster hébergé

Pour accélérer le processus de sauvegarde d'un cluster hébergé et sa restauration dans la même région sur AWS, vous pouvez modifier et exécuter un script.

Procédure

1. Remplacez les variables du script suivant par vos propres informations :

```
# Fill the Common variables to fit your environment, this is just a sample
SSH_KEY_FILE=${HOME}/.ssh/id_rsa.pub
BASE_PATH=${HOME}/hypershift
BASE_DOMAIN="aws.sample.com"
PULL_SECRET_FILE="${HOME}/pull_secret.json"
AWS_CREDS="${HOME}/.aws/credentials"
CONTROL_PLANE_AVAILABILITY_POLICY=SingleReplica
HYPERSHIFT_PATH=${BASE_PATH}/src/hypershift
HYPERSHIFT_CLI=${HYPERSHIFT_PATH}/bin/hypershift
HYPERSHIFT_IMAGE=${HYPERSHIFT_IMAGE:-"quay.io/${USER}/hypershift:latest"}
NODE_POOL_REPLICAS=${NODE_POOL_REPLICAS:-2}

# MGMT Context
MGMT_REGION=us-west-1
MGMT_CLUSTER_NAME="${USER}-dev"
MGMT_CLUSTER_NS=${USER}
MGMT_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${MGMT_CLUSTER_NS}-
${MGMT_CLUSTER_NAME}"
MGMT_KUBECONFIG="${MGMT_CLUSTER_DIR}/kubeconfig"

# MGMT2 Context
MGMT2_CLUSTER_NAME="${USER}-dest"
MGMT2_CLUSTER_NS=${USER}
MGMT2_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${MGMT2_CLUSTER_NS}-
${MGMT2_CLUSTER_NAME}"
MGMT2_KUBECONFIG="${MGMT2_CLUSTER_DIR}/kubeconfig"

# Hosted Cluster Context
HC_CLUSTER_NS=clusters
HC_REGION=us-west-1
HC_CLUSTER_NAME="${USER}-hosted"
HC_CLUSTER_DIR="${BASE_PATH}/hosted_clusters/${HC_CLUSTER_NS}-
${HC_CLUSTER_NAME}"
HC_KUBECONFIG="${HC_CLUSTER_DIR}/kubeconfig"
```

```
BACKUP_DIR=${HC_CLUSTER_DIR}/backup

BUCKET_NAME="${USER}-hosted-${MGMT_REGION}"

# DNS
AWS_ZONE_ID="Z026552815SS3YPH9H6MG"
EXTERNAL_DNS_DOMAIN="guest.jpdv.aws.kerbeross.com"
```

2. Enregistrez le script dans votre système de fichiers local.

3. Exécutez le script en entrant la commande suivante :

```
source <env_file>
```

où : **env_file** est le nom du fichier dans lequel vous avez enregistré le script.