OpenShift Dedicated 4

Le stockage

Configuration du stockage pour OpenShift clusters dédiés

Last Updated: 2025-05-27

OpenShift Dedicated 4 Le stockage

Configuration du stockage pour OpenShift clusters dédiés

Notice légale

Résumé

Ce document fournit des informations sur la configuration du stockage pour les clusters dédiés OpenShift.

Table des matières

CHAPITRE 1. APERÇU DU STOCKAGE DÉDIÉ À OPENSHIFT

Le logiciel OpenShift Dedicated prend en charge plusieurs types de stockage, tant pour les fournisseurs sur site que pour les fournisseurs de cloud. Dans un cluster dédié OpenShift, vous pouvez gérer le stockage de conteneurs pour des données persistantes et non persistantes.

1.1. GLOSSAIRE DES TERMES COMMUNS POUR OPENSHIFT STOCKAGE DÉDIÉ

Ce glossaire définit des termes communs qui sont utilisés dans le contenu de stockage.

Les modes d'accès

Les modes d'accès au volume décrivent les capacités de volume. Il est possible d'utiliser des modes d'accès pour correspondre à la revendication de volume persistant (PVC) et au volume persistant (PV). Ce qui suit sont les exemples de modes d'accès:

- ReadWriteOnce (RWO)
- LectureOnlyMany (ROX)
- ReadWriteMany (RWX)
- ReadWriteOncePod (RWOP)

Configuration de la carte

La carte de configuration fournit un moyen d'injecter des données de configuration dans des pods. Les données stockées dans une carte de configuration peuvent être référencées dans un volume de type ConfigMap. Les applications qui s'exécutent dans un pod peuvent utiliser ces données.

Interface de stockage de conteneurs (CSI)

La spécification API pour la gestion du stockage des conteneurs dans différents systèmes d'orchestration de conteneurs (CO).

Disposition dynamique

Le framework vous permet de créer des volumes de stockage à la demande, éliminant ainsi la nécessité pour les administrateurs de cluster de pré-fournir le stockage persistant.

Le stockage éphémère

Les pods et les conteneurs peuvent nécessiter un stockage local temporaire ou transitoire pour leur fonctionnement. La durée de vie de ce stockage éphémère ne s'étend pas au-delà de la durée de vie de la gousse individuelle, et ce stockage éphémère ne peut pas être partagé entre les gousses.

fsGroup

Le groupe fsGroup définit un identifiant de groupe de système de fichiers d'un pod.

HostPath

Le volume hostPath d'un cluster OpenShift Container Platform monte un fichier ou un répertoire à partir du système de fichiers du nœud hôte dans votre pod.

Clé KMS

Le Key Management Service (KMS) vous aide à atteindre le niveau de cryptage requis de vos données sur différents services. vous pouvez utiliser la clé KMS pour chiffrer, déchiffrer et recrypter les données.

Les volumes locaux

Le volume local représente un périphérique de stockage local monté tel qu'un disque, une partition ou un répertoire.

Fondation de données OpenShift

Fournisseur de stockage agnostique persistant pour OpenShift Container Platform prenant en charge le fichier, le blocage et le stockage d'objets, que ce soit en interne ou dans des clouds hybrides

Le stockage persistant

Les pods et les conteneurs peuvent nécessiter un stockage permanent pour leur fonctionnement. Le logiciel OpenShift Dedicated utilise le framework de volume persistant de Kubernetes (PV) pour permettre aux administrateurs de cluster de fournir un stockage persistant pour un cluster. Les développeurs peuvent utiliser le PVC pour demander des ressources PV sans avoir une connaissance spécifique de l'infrastructure de stockage sous-jacente.

Les volumes persistants (PV)

Le logiciel OpenShift Dedicated utilise le framework de volume persistant de Kubernetes (PV) pour permettre aux administrateurs de cluster de fournir un stockage persistant pour un cluster. Les développeurs peuvent utiliser le PVC pour demander des ressources PV sans avoir une connaissance spécifique de l'infrastructure de stockage sous-jacente.

Allégations de volume persistant (PVC)

Il est possible d'utiliser un PVC pour monter un Volume Persistent dans un Pod. Il est possible d'accéder au stockage sans connaître les détails de l'environnement cloud.

La pod

Il y a un ou plusieurs conteneurs avec des ressources partagées, telles que des adresses de volume et IP, qui s'exécutent dans votre cluster dédié OpenShift. Le pod est la plus petite unité de calcul définie, déployée et gérée.

La politique de récupération

Il s'agit d'une politique qui indique au cluster ce qu'il faut faire avec le volume après sa publication. La politique de récupération d'un volume peut être Retain, Recycle ou Supprimer.

Contrôle d'accès basé sur le rôle (RBAC)

Le contrôle d'accès basé sur les rôles (RBAC) est une méthode de régulation de l'accès aux ressources informatiques ou réseau en fonction des rôles des utilisateurs individuels au sein de votre organisation.

Applications apatrides

L'application apatride est un programme d'application qui n'enregistre pas les données client générées dans une session pour une utilisation dans la prochaine session avec ce client.

Applications étatiques

L'application étatique est un programme d'application qui enregistre les données sur un stockage persistant sur disque. Le serveur, le client et les applications peuvent utiliser un stockage de disque persistant. Dans OpenShift Dedicated, vous pouvez utiliser l'objet Statefulset pour gérer le déploiement et la mise à l'échelle d'un ensemble de Pods, et vous garantit la commande et l'unicité de ces Pods.

Approvisionnement statique

L'administrateur de cluster crée un certain nombre de PV. Les PVS contiennent les détails du stockage. Les PVS existent dans l'API Kubernetes et sont disponibles pour la consommation.

Le stockage

Le logiciel OpenShift Dedicated prend en charge de nombreux types de stockage, tant pour les fournisseurs sur site que pour les fournisseurs de cloud. Dans un cluster dédié OpenShift, vous pouvez gérer le stockage de conteneurs pour des données persistantes et non persistantes.

Classe de stockage

La classe de stockage permet aux administrateurs de décrire les classes de stockage qu'ils offrent. Différentes classes peuvent correspondre à la qualité des niveaux de service, aux politiques de sauvegarde, aux politiques arbitraires déterminées par les administrateurs de clusters.

1.2. LES TYPES DE STOCKAGE

Le stockage dédié OpenShift est largement classé en deux catégories, à savoir le stockage éphémère et le stockage persistant.

1.2.1. Le stockage éphémère

Les pods et les conteneurs sont éphémères ou transitoires et sont conçus pour des applications apatrides. Le stockage éphémère permet aux administrateurs et aux développeurs de mieux gérer le stockage local pour certaines de leurs opérations. En savoir plus sur l'aperçu, les types et la gestion du stockage éphémère, voir Comprendre le stockage éphémère.

1.2.2. Le stockage persistant

Les applications étatiques déployées dans des conteneurs nécessitent un stockage persistant. Le logiciel OpenShift Dedicated utilise un cadre de stockage pré-prévisé appelé volumes persistants (PV) pour permettre aux administrateurs de cluster de fournir un stockage persistant. Les données à l'intérieur de ces volumes peuvent exister au-delà du cycle de vie d'une gousse individuelle. Les développeurs peuvent utiliser des claims de volume persistants (PVC) pour demander des exigences de stockage. En savoir plus sur l'aperçu, la configuration et le cycle de vie du stockage persistant, voir Comprendre le stockage persistant.

1.3. INTERFACE DE STOCKAGE DE CONTENEURS (CSI)

CSI est une spécification API pour la gestion du stockage des conteneurs dans différents systèmes d'orchestration de conteneurs (CO). Il est possible de gérer les volumes de stockage dans les environnements natifs des conteneurs, sans avoir une connaissance spécifique de l'infrastructure de stockage sous-jacente. Avec le CSI, le stockage fonctionne uniformément sur différents systèmes d'orchestration de conteneurs, quels que soient les fournisseurs de stockage que vous utilisez. En savoir plus sur CSI, voir Utiliser l'interface de stockage de conteneurs (CSI).

1.4. DISPOSITION DYNAMIQUE

Dynamic Provisioning vous permet de créer des volumes de stockage à la demande, éliminant ainsi la nécessité pour les administrateurs de cluster de pré-fournir le stockage. Consultez Dynamic provisioning pour plus d'informations sur le provisionnement dynamique.

CHAPITRE 2. COMPRENDRE LE STOCKAGE ÉPHÉMÈRE

2.1. APERÇU GÉNÉRAL

En plus du stockage persistant, les pods et les conteneurs peuvent nécessiter un stockage local éphémère ou transitoire pour leur fonctionnement. La durée de vie de ce stockage éphémère ne s'étend pas au-delà de la durée de vie de la gousse individuelle, et ce stockage éphémère ne peut pas être partagé entre les gousses.

Les gousses utilisent un stockage local éphémère pour l'espace de grattage, la mise en cache et les journaux. Les questions liées à l'absence de comptabilité locale de stockage et d'isolement sont les suivantes:

- Les pods ne peuvent pas détecter la quantité de stockage local disponible pour eux.
- Les pods ne peuvent pas demander un stockage local garanti.
- Le stockage local est une ressource de meilleur effort.
- Les gousses peuvent être expulsées en raison d'autres gousses remplissant le stockage local, après quoi de nouvelles gousses ne sont pas admises tant que le stockage n'est pas suffisant.

Contrairement aux volumes persistants, le stockage éphémère n'est pas structuré et l'espace est partagé entre tous les pods fonctionnant sur un nœud, en plus d'autres utilisations par le système, le temps d'exécution du conteneur et OpenShift Dedicated. Le cadre de stockage éphémère permet aux pods de spécifier leurs besoins de stockage locaux transitoires. Il permet également à OpenShift Dedicated de programmer les pods le cas échéant et de protéger le nœud contre l'utilisation excessive du stockage local.

Alors que le cadre de stockage éphémère permet aux administrateurs et aux développeurs de mieux gérer le stockage local, le débit et la latence d'E/S ne sont pas directement effectués.

2.2. LES TYPES DE STOCKAGE ÉPHÉMÈRE

Le stockage local éphémère est toujours disponible dans la partition principale. Il existe deux façons fondamentales de créer la partition principale: root et runtime.

La racine
Cette partition contient le répertoire racine kubelet, /var/lib/kubelet/ par défaut, et le répertoire /var/log/. Cette partition peut être partagée entre les pods utilisateur, l'OS et les démons système Kubernetes. Cette partition peut être consommée par des pods à travers des volumes EmptyDir, des journaux de conteneurs, des couches d'image et des couches écrivant des conteneurs. Kubelet gère l'accès partagé et l'isolement de cette partition. Cette partition est éphémère, et les applications ne peuvent s'attendre à aucun SLA de performance, tel que le disque IOPS, de cette partition.

Durée d'exécution
Il s'agit d'une partition optionnelle que les runtimes peuvent utiliser pour les systèmes de fichiers superposés. Les tentatives d'OpenShift Dedicated d'identifier et de fournir un accès partagé ainsi que l'isolement à cette partition. Les calques d'image de conteneur et les calques writables sont stockés ici. Dans le cas où la partition d'exécution existe, la partition racine ne contient aucune couche d'image ou autre stockage writable.

2.3. GESTION DU STOCKAGE ÉPHÉMÈRE

Les administrateurs de clusters peuvent gérer le stockage éphémère au sein d'un projet en définissant des quotas qui définissent les plages limites et le nombre de demandes de stockage éphémère dans tous les pods dans un état non terminal. Les développeurs peuvent également définir des requêtes et des limites sur cette ressource de calcul au niveau du pod et du conteneur.

Il est possible de gérer le stockage éphémère local en spécifiant les requêtes et les limites. Chaque conteneur dans une gousse peut spécifier ce qui suit:

- **contient spec.resources.limits.ephemeral-stockage**
- **contient spec.resources.requests.ephemeral-stockage**

2.3.1. Limites de stockage éphémères et unités de demande

Les limites et les demandes de stockage éphémère sont mesurées en quantités d'octets. Il est possible d'exprimer le stockage en entier simple ou en tant que numéro de point fixe en utilisant l'un de ces suffixes: E, P, T, G, M, k. Il est également possible d'utiliser la puissance de deux équivalents: Ei, Pi, Ti, Gi, Mi, Ki.

Ainsi, les quantités suivantes représentent à peu près la même valeur : 128974848, 129e6, 129M et 123Mi.

IMPORTANT

Les suffixes pour chaque quantité d'octets sont sensibles à la casse. Assurez-vous d'utiliser le bon cas. Employez le "M" sensible à la casse, tel que utilisé dans "400M" pour régler la demande à 400 mégaoctets. Faites appel au "400Mi" sensible à la casse pour demander 400 mébioctets. Lorsque vous spécifiez "400m" de stockage éphémère, les demandes de stockage ne sont que de 0,4 octets.

2.3.2. Exemples de demandes de stockage éphémères et limites

Le fichier de configuration d'exemple suivant affiche un pod avec deux conteneurs:

- Chaque conteneur demande 2GiB de stockage éphémère local.
- Chaque conteneur a une limite de 4GiB de stockage éphémère local.
- Au niveau de la gousse, kubelet établit une limite globale de stockage des gousses en ajoutant les limites de tous les contenants de cette gousse.
    - Dans ce cas, l'utilisation totale de stockage au niveau de la pod est la somme de l'utilisation du disque de tous les conteneurs plus les volumes vides du pod.
    - Le pod a donc une demande de 4GiB de stockage éphémère local, et une limite de 8GiB de stockage éphémère local.

Exemple de configuration de stockage éphémère avec quotas et limites

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
```

```
    image: images.my-company.example/app:v4
    resources:
      requests:
        ephemeral-storage: "2Gi" (1)
      limits:
        ephemeral-storage: "4Gi" (2)
    volumeMounts:
    - name: ephemeral
      mountPath: "/tmp"
  - name: log-aggregator
    image: images.my-company.example/log-aggregator:v6
    resources:
      requests:
        ephemeral-storage: "2Gi"
      limits:
        ephemeral-storage: "4Gi"
    volumeMounts:
    - name: ephemeral
      mountPath: "/tmp"
  volumes:
    - name: ephemeral
      emptyDir: {}
```

1. Demande de conteneur pour le stockage éphémère local.
2. Limite de conteneur pour le stockage éphémère local.

2.3.3. Effets de configuration de stockage éphémère de la planification et de l'expulsion des pods

Les paramètres dans la spécification de pod affectent à la fois la façon dont le planificateur prend une décision sur la planification des pods et quand kubelet expulse des pods.

- D'abord, le planificateur veille à ce que la somme des demandes de ressources des conteneurs programmés soit inférieure à la capacité du nœud. Dans ce cas, le pod ne peut être affecté à un nœud que si le stockage éphémère disponible du nœud (ressourceallocatable) est supérieur à 4GiB.

- Deuxièmement, au niveau du conteneur, parce que le premier conteneur fixe une limite de ressources, le gestionnaire d'expulsion de kubelet mesure l'utilisation du disque de ce conteneur et expulse le pod si l'utilisation du conteneur dépasse sa limite (4GiB). Le gestionnaire d'éviction de kubelet marque également le pod d'expulsion si l'utilisation totale dépasse la limite globale de stockage des pod (8GiB).

2.4. CONTRÔLE DU STOCKAGE ÉPHÉMÈRE

Il est possible d'utiliser /bin/df comme outil pour surveiller l'utilisation du stockage éphémère sur le volume où se trouvent les données du conteneur éphémère, qui est /var/lib/kubelet et /var/lib/conteneurs. L'espace disponible uniquement pour /var/lib/kubelet est affiché lorsque vous utilisez la commande df si /var/lib/containers est placé sur un disque séparé par l'administrateur du cluster.

Afin d'afficher les valeurs lisibles par l'homme de l'espace utilisé et disponible dans /var/lib, entrez la commande suivante:

```
$ df -h /var/lib
```

La sortie montre l'utilisation du stockage éphémère dans /var/lib:

Exemple de sortie

```
Filesystem  Size  Used Avail Use% Mounted on
/dev/disk/by-partuuid/4cd1448a-01    69G   32G   34G  49% /
```

CHAPITRE 3. COMPRENDRE LE STOCKAGE PERSISTANT

3.1. APERÇU DU STOCKAGE PERSISTANT

La gestion du stockage est un problème distinct de la gestion des ressources de calcul. Le logiciel OpenShift Dedicated utilise le framework de volume persistant de Kubernetes (PV) pour permettre aux administrateurs de cluster de fournir un stockage persistant pour un cluster. Les développeurs peuvent utiliser des claims de volume persistants (PVC) pour demander des ressources PV sans avoir une connaissance spécifique de l'infrastructure de stockage sous-jacente.

Les PVC sont spécifiques à un projet, et sont créés et utilisés par les développeurs comme un moyen d'utiliser un PV. Les ressources photovoltaïques par elles-mêmes ne sont soumises à aucun projet; elles peuvent être partagées sur l'ensemble du cluster dédié OpenShift et revendiquées de n'importe quel projet. Après qu'un PV soit lié à un PVC, le PV ne peut alors pas être lié à des PVC supplémentaires. Cela a pour effet de fixer un PV lié à un seul espace de noms, celui du projet de liaison.

Les PVS sont définis par un objet API PersistentVolume, qui représente un morceau de stockage existant dans le cluster qui a été soit statiquement provisionné par l'administrateur du cluster, soit dynamiquement provisionné à l'aide d'un objet StorageClass. C'est une ressource dans le cluster tout comme un nœud est une ressource de cluster.

Les PVS sont des plugins de volume comme les volumes, mais ont un cycle de vie indépendant de toute dose individuelle qui utilise le PV. Les objets PV capturent les détails de la mise en œuvre du stockage, que ce soit NFS, iSCSI, ou un système de stockage spécifique au cloud provider.

IMPORTANT

La disponibilité élevée de stockage dans l'infrastructure est laissée au fournisseur de stockage sous-jacent.

Les PVC sont définis par un objet API PersistentVolumeClaim, qui représente une demande de stockage par un développeur. Il est similaire à un pod en ce que les gousses consomment des ressources de nœud et que les PVC consomment des ressources PV. À titre d'exemple, les pods peuvent demander des niveaux de ressources spécifiques, tels que le CPU et la mémoire, tandis que les PVC peuvent demander une capacité de stockage spécifique et des modes d'accès. Ils peuvent par exemple être montés une fois lecture-écriture ou plusieurs fois en lecture seule.

3.2. CYCLE DE VIE D'UN VOLUME ET D'UNE REVENDICATION

Les PVS sont des ressources dans le cluster. Les PVC sont des demandes pour ces ressources et agissent également comme des vérifications de réclamation de la ressource. L'interaction entre les PV et les PVC a le cycle de vie suivant.

3.2.1. Entreposage des provisions

En réponse aux demandes d'un développeur défini dans un PVC, un administrateur de cluster configure un ou plusieurs provisionneurs dynamiques qui fournissent un stockage et un PV correspondant.

3.2.2. Lier les revendications

Lorsque vous créez un PVC, vous demandez une quantité spécifique de stockage, spécifiez le mode d'accès requis et créez une classe de stockage pour décrire et classer le stockage. La boucle de commande dans les montres principales pour les nouveaux PVC et lie le nouveau PVC à un PV

approprié. En cas d'absence d'un PV approprié, un provisionneur pour la classe de stockage en crée un.

La taille de tous les PV pourrait dépasser votre taille de PVC. Cela est particulièrement vrai avec les PV fournis manuellement. Afin de minimiser l'excès, OpenShift Dedicated se lie au plus petit PV qui correspond à tous les autres critères.

Les réclamations restent illimitées si un volume correspondant n'existe pas ou ne peut pas être créé avec un fournisseur disponible desservant une classe de stockage. Les revendications sont liées au fur et à mesure que les volumes correspondants deviennent disponibles. À titre d'exemple, un cluster avec de nombreux volumes 50Gi fournis manuellement ne correspondrait pas à un PVC demandant 100Gi. Le PVC peut être lié lorsqu'un PV 100Gi est ajouté au cluster.

3.2.3. Employez des gousses et des PV revendiqués

Les pods utilisent les revendications comme volumes. Le cluster inspecte la revendication pour trouver le volume lié et monte ce volume pour un pod. Dans le cas des volumes qui prennent en charge plusieurs modes d'accès, vous devez spécifier quel mode s'applique lorsque vous utilisez la revendication comme volume dans un pod.

Lorsque vous avez une réclamation et que cette revendication est liée, le PV lié vous appartient aussi longtemps que vous en avez besoin. Il est possible de programmer des pods et d'accéder aux PV revendiqués en incluant persistantVolumeClaim dans le bloc volumes du pod.

NOTE

Lorsque vous attachez des volumes persistants qui ont un nombre élevé de fichiers à des pods, ces gousses peuvent échouer ou peuvent prendre beaucoup de temps pour démarrer. Lorsque vous utilisez des volumes persistants avec des nombres de fichiers élevés dans OpenShift, pourquoi les pods ne commencent-ils pas ou prennent-ils trop de temps pour atteindre l'état « Ready »?

3.2.4. Libérer un volume persistant

Lorsque vous avez terminé avec un volume, vous pouvez supprimer l'objet PVC de l'API, ce qui permet la récupération de la ressource. Le volume est considéré comme libéré lorsque la revendication est supprimée, mais il n'est pas encore disponible pour une autre revendication. Les données du demandeur précédent restent sur le volume et doivent être traitées conformément à la politique.

3.2.5. La politique de récupération des volumes persistants

La politique de récupération d'un volume persistant indique au cluster ce qu'il faut faire avec le volume après sa sortie. La politique de récupération d'un volume peut être Retain, Recycle ou Supprimer.

- La stratégie de récupération permet la récupération manuelle de la ressource pour les plugins de volume qui la prennent en charge.
- La politique de recyclage recycle le volume dans le bassin de volumes persistants non liés une fois qu'il est libéré de sa réclamation.

IMPORTANT

La politique de récupération de recyclage est obsolète dans OpenShift Dedicated 4. Le provisionnement dynamique est recommandé pour une fonctionnalité équivalente et améliorée.

- La stratégie de récupération supprime à la fois l'objet PersistentVolume d'OpenShift Dedicated et l'actif de stockage associé dans l'infrastructure externe, comme Amazon Elastic Block Store (Amazon EBS) ou VMware vSphere.

NOTE

Les volumes provisionnés dynamiquement sont toujours supprimés.

3.2.6. La récupération d'un volume persistant manuellement

Lorsqu'une revendication de volume persistant (PVC) est supprimée, le volume persistant (PV) existe toujours et est considéré comme « libéré ». Cependant, le PV n'est pas encore disponible pour une autre demande parce que les données du précédent demandeur restent sur le volume.

Procédure

De récupérer manuellement le PV en tant qu'administrateur de cluster:

1. Efface le PV.

```
$ oc delete pv <pv-name>
```

   L'actif de stockage associé dans l'infrastructure externe, tel qu'un volume AWS EBS ou GCE PD, existe toujours après la suppression du PV.

2. Nettoyer les données sur l'actif de stockage associé.

3. Effacer l'actif de stockage associé. Alternativement, pour réutiliser le même actif de stockage, créez un nouveau PV avec la définition de l'actif de stockage.

Le PV récupéré est maintenant disponible pour une utilisation par un autre PVC.

3.2.7. Changer la politique de récupération d'un volume persistant

Changer la politique de récupération d'un volume persistant:

1. Énumérez les volumes persistants de votre cluster:

```
$ oc get pv
```

   #### Exemple de sortie

```
NAME                                       CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS
CLAIM           STORAGECLASS     REASON    AGE
 pvc-b6efd8da-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Delete          Bound
default/claim1    manual                    10s
 pvc-b95650f8-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Delete          Bound
default/claim2    manual                    6s
 pvc-bb3ca71d-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Delete          Bound
default/claim3    manual                    3s
```

2. Choisissez un de vos volumes persistants et modifiez sa politique de récupération:

```
$ oc patch pv <your-pv-name> -p '{"spec":{"persistentVolumeReclaimPolicy":"Retain"}}'
```

3. Assurez-vous que votre volume persistant choisi a la bonne politique:

```
$ oc get pv
```

Exemple de sortie

```
NAME                                       CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS
CLAIM           STORAGECLASS     REASON   AGE
 pvc-b6efd8da-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Delete          Bound
default/claim1    manual                   10s
 pvc-b95650f8-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Delete          Bound
default/claim2    manual                   6s
 pvc-bb3ca71d-b7b5-11e6-9d58-0ed433a7dd94   4Gi        RWO           Retain          Bound
default/claim3    manual                   3s
```

Dans la sortie précédente, le volume lié à la réclamation par défaut3 a maintenant une politique de récupération de retenue. Le volume ne sera pas automatiquement supprimé lorsqu'un utilisateur supprime la réclamation par défaut/claim3.

3.3. LES VOLUMES PERSISTANTS

Chaque PV contient une spécification et un statut, qui est la spécification et l'état du volume, par exemple:

Exemple de définition d'objet PersistentVolume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001 1
spec:
  capacity:
    storage: 5Gi 2
  accessModes:
    - ReadWriteOnce 3
  persistentVolumeReclaimPolicy: Retain 4
  ...
status:
  ...
```

1. Le nom du volume persistant.
2. La quantité de stockage disponible pour le volume.
3. Le mode d'accès, définissant les autorisations de lecture-écriture et de montage.
4. La politique de récupération, indiquant comment la ressource devrait être gérée une fois qu'elle est publiée.

Il est possible d'afficher le nom d'un PVC lié à un PV en exécutant la commande suivante:

```
$ oc get pv <pv-name> -o jsonpath='{.spec.claimRef.name}'
```

3.3.1. Les types de PV

Le logiciel OpenShift Dedicated prend en charge les plugins de volume persistants suivants:

- AWS Elastic Block Store (EBS)
- Disque persistant GCP
- GCP Filestore

3.3.2. Capacité

Généralement, un volume persistant (PV) a une capacité de stockage spécifique. Ceci est défini en utilisant l'attribut de capacité du PV.

Actuellement, la capacité de stockage est la seule ressource pouvant être définie ou demandée. Les attributs futurs peuvent inclure IOPS, le débit, et ainsi de suite.

3.3.3. Les modes d'accès

Le volume persistant peut être monté sur un hôte de quelque manière que ce soit pris en charge par le fournisseur de ressources. Les fournisseurs ont des capacités différentes et les modes d'accès de chaque PV sont définis sur les modes spécifiques pris en charge par ce volume particulier. À titre d'exemple, NFS peut prendre en charge plusieurs clients en lecture-écriture, mais un NFS PV spécifique peut être exporté sur le serveur en lecture seule. Chaque PV obtient son propre ensemble de modes d'accès décrivant les capacités spécifiques de ce PV.

Les revendications sont appariées à des volumes avec des modes d'accès similaires. Les deux seuls critères correspondants sont les modes d'accès et la taille. Les modes d'accès d'une revendication représentent une demande. Donc, vous pourriez être accordé plus, mais jamais moins. À titre d'exemple, si une revendication demande RWO, mais le seul volume disponible est un NFS PV (RWO+ROX+RWX), la revendication correspondrait alors à NFS parce qu'elle supporte RWO.

Les matchs directs sont toujours tentés en premier. Les modes du volume doivent correspondre ou contenir plus de modes que vous ne l'avez demandé. La taille doit être supérieure ou égale à ce qui est attendu. Lorsque deux types de volumes, tels que NFS et iSCSI, ont le même ensemble de modes d'accès, l'un d'eux peut correspondre à une revendication avec ces modes. Il n'y a pas de commande entre les types de volumes et aucun moyen de choisir un type par rapport à un autre.

Les volumes avec les mêmes modes sont regroupés, puis triés par taille, les plus petits à les plus grands. Le liant obtient le groupe avec des modes assortis et itère sur chacun, dans l'ordre de taille, jusqu'à ce qu'une taille corresponde.

IMPORTANT

Les modes d'accès au volume décrivent les capacités de volume. Ce ne sont pas des contraintes forcées. Le fournisseur de stockage est responsable des erreurs d'exécution résultant de l'utilisation invalide de la ressource. Les erreurs dans le fournisseur apparaissent au moment de l'exécution sous forme d'erreurs de montage.

Le tableau suivant répertorie les modes d'accès:

Tableau 3.1. Les modes d'accès

Le mode d'accès	Abréviation de CLI	Description
À propos de ReadWriteOnce	**LE RWO**	Le volume peut être monté en lecture-écriture par un seul nœud.
ReadWriteOncePod [1]	**LE RWOP**	Le volume peut être monté en lecture-écriture par un seul pod sur un seul nœud.

1. Le RWOP utilise la fonction de montage SELinux. Cette fonctionnalité est dépendante du pilote et activée par défaut dans ODF, AWS EBS, Azure Disk, GCP PD, IBM Cloud Block Storage volume, Cinder et vSphere. En ce qui concerne les conducteurs tiers, veuillez contacter votre fournisseur de stockage.

Tableau 3.2. Les modes d'accès pris en charge pour les volumes persistants

Plugin de volume	L'histoire de ReadWriteOnce [1]	ReadWriteOncePod	LireOnlyMany	ReadWriteMany
AWS EBS [2]	▮	▮		
AWS EFS	▮	▮	▮	▮
Disque persistant GCP	▮	▮		
GCP Filestore	▮	▮	▮	▮
Le stockage LVM	▮	▮		

1. Les volumes ReadWriteOnce (RWO) ne peuvent pas être montés sur plusieurs nœuds. En cas de défaillance d'un nœud, le système ne permet pas que le volume RWO attaché soit monté sur un nouveau nœud car il est déjà affecté au nœud défaillant. Lorsque vous rencontrez un message d'erreur multi-attaché en conséquence, forcez à supprimer le pod sur un nœud d'arrêt ou un nœud planté pour éviter la perte de données dans les charges de travail critiques, par exemple lorsque des volumes persistants dynamiques sont attachés.

2. Employez une stratégie de déploiement recréée pour les pods qui s'appuient sur AWS EBS.

3. Les volumes de blocs bruts prennent en charge le mode d'accès ReadWriteMany (RWX) pour Fibre Channel et iSCSI. En savoir plus, voir « Support de volume de verrouillage ».

3.3.4. De la phase

Les volumes peuvent être trouvés dans l'une des phases suivantes:

Tableau 3.3. Les phases de volume

De la phase	Description
Disponible	C'est une ressource gratuite qui n'est pas encore liée à une réclamation.
Lié	Le volume est lié à une réclamation.
Libéré	La réclamation a été supprimée, mais la ressource n'est pas encore récupérée par le cluster.
J'ai échoué	Le volume a échoué à sa récupération automatique.

3.3.4.1. Dernière phase de transition

Le champ LastPhaseTransitionTime dispose d'un horodatage qui met à jour chaque fois qu'un volume persistant (PV) passe à une phase différente (pv.Status.Phase). Afin de trouver l'heure de la dernière transition de phase pour un PV, exécutez la commande suivante:

```
$ oc get pv <pv-name> -o json | jq '.status.lastPhaseTransitionTime' 1
```

1 Indiquez le nom du PV que vous souhaitez voir la dernière phase de transition.

3.3.4.2. Les options de montage

Lors du montage d'un PV, vous pouvez spécifier les options de montage à l'aide de l'attribut MountOptions.

À titre d'exemple:

Exemple d'options de montage

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  mountOptions: 1
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
  persistentVolumeReclaimPolicy: Retain
  claimRef:
    name: claim1
    namespace: default
```

1 Des options de montage spécifiées sont utilisées lors du montage du PV sur le disque.

Les types PV suivants prennent en charge les options de montage:

- AWS Elastic Block Store (EBS)
- Disque persistant GCE

3.4. ALLÉGATIONS RELATIVES AU VOLUME PERSISTANT

Chaque objet PersistentVolumeClaim contient une spécification et un statut, qui est la spécification et le statut de la revendication de volume persistant (PVC), par exemple:

Exemple de définition d'objet PersistentVolumeClaim

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclaim 1
spec:
  accessModes:
    - ReadWriteOnce 2
  resources:
    requests:
      storage: 8Gi 3
  storageClassName: gold 4
status:
  ...
```

1. Le nom du PVC.
2. Le mode d'accès, définissant les autorisations de lecture-écriture et de montage.
3. La quantité de stockage disponible pour le PVC.
4. Le nom de la classe de stockage requise par la revendication.

3.4.1. Classes de stockage

Les revendications peuvent éventuellement demander une classe de stockage spécifique en spécifiant le nom de la classe de stockage dans l'attribut StorageClassName. Il n'y a que les PV de la classe demandée, ceux qui ont la même classe de stockage que le PVC, peuvent être liés au PVC. L'administrateur du cluster peut configurer des provisionneurs dynamiques pour desservir une ou plusieurs classes de stockage. L'administrateur du cluster peut créer un PV à la demande qui correspond aux spécifications du PVC.

IMPORTANT

L'opérateur de stockage du cluster peut installer une classe de stockage par défaut en fonction de la plate-forme utilisée. Cette classe de stockage est détenue et contrôlée par l'opérateur. Il ne peut pas être supprimé ou modifié au-delà de la définition des annotations et des étiquettes. Lorsqu'un comportement différent est souhaité, vous devez définir une classe de stockage personnalisée.

L'administrateur du cluster peut également définir une classe de stockage par défaut pour tous les PVC.

Lorsqu'une classe de stockage par défaut est configurée, le PVC doit explicitement demander que les annotations StorageClass ou StorageClassName définies sur «» soient liées à un PV sans classe de stockage.

NOTE

Lorsque plus d'une classe de stockage est marquée par défaut, un PVC ne peut être créé que si la classe de stockage est explicitement spécifiée. Donc, une seule classe de stockage doit être définie par défaut.

3.4.2. Les modes d'accès

Les revendications utilisent les mêmes conventions que les volumes lors de la demande de stockage avec des modes d'accès spécifiques.

3.4.3. Ressources

Les revendications, telles que les pods, peuvent demander des quantités spécifiques d'une ressource. Dans ce cas, la demande est de stockage. Le même modèle de ressource s'applique aux volumes et aux revendications.

3.4.4. Créances en tant que volumes

Les pods accèdent au stockage en utilisant la revendication comme volume. Les revendications doivent exister dans le même espace de noms que le pod utilisant la revendication. Le cluster trouve la revendication dans l'espace de noms du pod et l'utilise pour obtenir le PersistentVolume à l'appui de la revendication. Le volume est monté sur l'hôte et dans le pod, par exemple:

Monter le volume vers l'hôte et dans l'exemple du pod

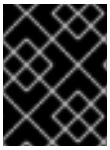
```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: myfrontend
      image: dockerfile/nginx
      volumeMounts:
      - mountPath: "/var/www/html" 1
        name: mypd 2
  volumes:
    - name: mypd
      persistentVolumeClaim:
        claimName: myclaim 3
```

1. Chemin pour monter le volume à l'intérieur de la gousse.
2. Le nom du volume à monter. Il ne faut pas monter sur la racine du conteneur, /, ou tout chemin qui est le même dans l'hôte et le conteneur. Cela peut corrompre votre système hôte si le conteneur est suffisamment privilégié, comme les fichiers hôte /dev/pts. Il est sûr de monter l'hôte en utilisant /host.
3. Le nom du PVC, qui existe dans le même espace de noms, à utiliser.

3.5. BLOC DE SUPPORT DE VOLUME

L'OpenShift Dedicated peut fournir statiquement des volumes de blocs bruts. Ces volumes n'ont pas de système de fichiers et peuvent fournir des avantages de performance pour les applications qui écrivent directement sur le disque ou implémentent leur propre service de stockage.

Les volumes de blocs bruts sont fournis en spécifiant volumeMode: Bloc dans la spécification PV et PVC.



IMPORTANT

Les pods utilisant des volumes de blocs bruts doivent être configurés pour permettre des conteneurs privilégiés.

Le tableau suivant affiche les plugins de volume supportant les volumes de blocs.

Tableau 3.4. Bloc de support de volume

Plugin de volume	Fourni manuellement	Approvisionnement dynamique	Entièrement pris en charge
Amazon Elastic Block Store (Amazon EBS)	À PROPOS DE ER	À PROPOS DE ER	À PROPOS DE ER
Amazon Elastic File Storage (Amazon EFS)			
GCP	À PROPOS DE ER	À PROPOS DE ER	À PROPOS DE ER
Le stockage LVM	À PROPOS DE ER	À PROPOS DE ER	À PROPOS DE ER

3.5.1. Exemples de volume de bloc

Exemple PV

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: block-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  volumeMode: Block 1
  persistentVolumeReclaimPolicy: Retain
  fc:
    targetWWNs: ["50060e801049cfd1"]
    lun: 0
    readOnly: false
```

1. le VolumeMode doit être réglé sur Block pour indiquer que ce PV est un volume de bloc brut.

Exemple de PVC

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: block-pvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Block (1)
  resources:
    requests:
      storage: 10Gi
```

1. le VolumeMode doit être réglé sur Block pour indiquer qu'un bloc brut de PVC est demandé.

Exemple de spécification de pod

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-block-volume
spec:
  containers:
    - name: fc-container
      image: fedora:26
      command: ["/bin/sh", "-c"]
      args: [ "tail -f /dev/null" ]
      volumeDevices: (1)
        - name: data
          devicePath: /dev/xvda (2)
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: block-pvc (3)
```

1. le volumeDevices, au lieu de volumeMounts, est utilisé pour bloquer les périphériques. Les sources PersistentVolumeClaim ne peuvent être utilisées qu'avec des volumes de blocs bruts.

2. DevicePath, au lieu de MountPath, représente le chemin vers l'appareil physique où le bloc brut est mappé au système.

3. La source de volume doit être de type persistantVolumeClaim et doit correspondre au nom du PVC comme prévu.

Tableau 3.5. Les valeurs acceptées pour volumeMode

La valeur	Défaut par défaut
Le système de fichiers	♪ oui ♪
Bloc	C) Non

Tableau 3.6. Des scénarios de liaison pour les volumes de blocs

Le volume PVMode	Le volume de PVCMode	Le résultat contraignant
Le système de fichiers	Le système de fichiers	Lier
Indéterminé	Indéterminé	Lier
Le système de fichiers	Indéterminé	Lier
Indéterminé	Le système de fichiers	Lier
Bloc	Bloc	Lier
Indéterminé	Bloc	Il n'y a pas de Bind
Bloc	Indéterminé	Il n'y a pas de Bind
Le système de fichiers	Bloc	Il n'y a pas de Bind
Bloc	Le système de fichiers	Il n'y a pas de Bind

IMPORTANT

Les valeurs non spécifiées entraînent la valeur par défaut de Filesystem.

3.6. À L'AIDE DE FSGROUP POUR RÉDUIRE LES TEMPS D'ATTENTE DES POD

Dans le cas où un volume de stockage contient de nombreux fichiers (~1 000 000 ou plus), vous pouvez connaître des délais de stockage.

Cela peut se produire parce que, par défaut, OpenShift Dedicated modifie récursivement la propriété et les autorisations pour que le contenu de chaque volume corresponde au fsGroup spécifié dans le contexte de sécurité d'un pod lorsque ce volume est monté. Dans le cas de grands volumes, la vérification et la modification de la propriété et des autorisations peuvent prendre du temps,

ralentissant le démarrage de la pod. Il est possible d'utiliser le champ fsGroupChangePolicy à l'intérieur d'un securityContext pour contrôler la manière dont OpenShift Dedicated vérifie et gère la propriété et les autorisations d'un volume.

fsGroupChangePolicy définit le comportement pour changer la propriété et l'autorisation du volume avant d'être exposé à l'intérieur d'un pod. Ce champ s'applique uniquement aux types de volume qui prennent en charge la propriété et les autorisations contrôlées par fsGroup. Ce champ a deux valeurs possibles:

- Changez uniquement les autorisations et la propriété si l'autorisation et la propriété du répertoire root ne correspondent pas aux autorisations attendues du volume. Cela peut aider à raccourcir le temps nécessaire pour changer la propriété et l'autorisation d'un volume pour réduire les délais de sortie des pod.
- Changez toujours l'autorisation et la propriété du volume lorsqu'un volume est monté.

exemple de la politique fsGroupChangePolicy

```
securityContext:
  runAsUser: 1000
  runAsGroup: 3000
  fsGroup: 2000
  fsGroupChangePolicy: "OnRootMismatch" 1
  ...
```

1 La fonction OnRootMismatch spécifie le changement d'autorisation récursif, ce qui permet d'éviter les problèmes de timeout de pod.

NOTE

FsGroupChangePolicyfield n'a aucun effet sur les types de volumes éphémères, tels que secret, configMap et vide.

CHAPITRE 4. CONFIGURATION DU STOCKAGE PERSISTANT

4.1. STOCKAGE PERSISTANT À L'AIDE D'AWS ELASTIC BLOCK STORE

Les clusters dédiés OpenShift sont préconçus avec quatre classes de stockage utilisant les volumes Amazon Elastic Block Store (Amazon EBS). Ces classes de stockage sont prêtes à être utilisées et une certaine familiarité avec Kubernetes et AWS est supposée.

Les quatre classes de stockage préconçues suivantes sont les suivantes:

Le nom	A) Prestataire
Gp2	Kubernetes.io/aws-ebs
gp2-csi	EBS.csi.aws.com
gp3 (par défaut)	Kubernetes.io/aws-ebs
gp3-csi	EBS.csi.aws.com

La classe de stockage gp3 est définie par défaut; cependant, vous pouvez sélectionner l'une des classes de stockage comme classe de stockage par défaut.

Le cadre de volume persistant de Kubernetes permet aux administrateurs de fournir un cluster avec un stockage persistant et donne aux utilisateurs un moyen de demander ces ressources sans avoir aucune connaissance de l'infrastructure sous-jacente. De manière dynamique, vous pouvez fournir des volumes Amazon EBS. Les volumes persistants ne sont pas liés à un seul projet ou espace de noms; ils peuvent être partagés sur le cluster OpenShift dédié. Les revendications de volume persistantes sont spécifiques à un projet ou à un espace de noms et peuvent être demandées par les utilisateurs. Il est possible de définir une clé KMS pour chiffrer des volumes persistants sur AWS. Les clusters nouvellement créés à l'aide de la version 4.10 d'OpenShift Dedicated utilisent par défaut le stockage gp3 et le pilote AWS EBS CSI.

IMPORTANT

La grande disponibilité du stockage dans l'infrastructure est laissée au fournisseur de stockage sous-jacent.

4.1.1. Création de la classe de stockage EBS

Les classes de stockage sont utilisées pour différencier et délimiter les niveaux et les usages de stockage. En définissant une classe de stockage, les utilisateurs peuvent obtenir des volumes persistants provisionnés dynamiquement.

4.1.2. Création de la revendication de volume persistante

Conditions préalables

Le stockage doit exister dans l'infrastructure sous-jacente avant de pouvoir être monté en tant que volume dans OpenShift Dedicated.

Procédure

1. Dans la console dédiée OpenShift, cliquez sur Stockage → Réclamations de volume persistant.
2. Dans l'aperçu des revendications de volume persistant, cliquez sur Créer une revendication de volume persistant.
3. Définissez les options souhaitées sur la page qui apparaît.
   a. Choisissez la classe de stockage précédemment créée dans le menu déroulant.
   b. Entrez un nom unique pour la revendication de stockage.
   c. Choisissez le mode d'accès. Cette sélection détermine l'accès en lecture et en écriture pour la revendication de stockage.
   d. Définir la taille de la revendication de stockage.
4. Cliquez sur Créer pour créer la revendication de volume persistante et générer un volume persistant.

4.1.3. Format du volume

Avant que OpenShift Dedicated ne monte le volume et le transmet à un conteneur, il vérifie que le volume contient un système de fichiers tel que spécifié par le paramètre fsType dans la définition de volume persistante. Lorsque l'appareil n'est pas formaté avec le système de fichiers, toutes les données de l'appareil sont effacées et l'appareil est automatiquement formaté avec le système de fichiers donné.

Cette vérification vous permet d'utiliser des volumes AWS non formatés comme volumes persistants, car OpenShift Dedicated les formate avant la première utilisation.

4.1.4. Le nombre maximal de volumes EBS sur un nœud

Par défaut, OpenShift Dedicated prend en charge un maximum de 39 volumes EBS attachés à un nœud. Cette limite est compatible avec les limites de volume AWS. La limite de volume dépend du type d'instance.

IMPORTANT

En tant qu'administrateur de cluster, vous devez utiliser les volumes de l'interface de stockage de conteneurs (CSI) et leurs classes de stockage respectives, mais jamais les deux types de volume en même temps. Le nombre maximum de volume EBS joint est compté séparément pour les volumes en arbre et CSI, ce qui signifie que vous pourriez avoir jusqu'à 39 volumes EBS de chaque type.

En ce qui concerne l'accès à des options de stockage supplémentaires, telles que des instantanés de volume, qui ne sont pas possibles avec les plug-ins de volume dans l'arbre, consultez AWS Elastic Block Store CSI Driver Operator.

4.1.5. Chiffrement des volumes persistants de conteneur sur AWS avec une clé KMS

La définition d'une clé KMS pour chiffrer des volumes persistants sur AWS est utile lorsque vous avez des directives de conformité et de sécurité explicites lors du déploiement sur AWS.

Conditions préalables

- L'infrastructure sous-jacente doit contenir le stockage.
- Il faut créer une clé KMS client sur AWS.

Procédure

1. Créer une classe de stockage:

```
$ cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <storage-class-name> 1
parameters:
  fsType: ext4 2
  encrypted: "true"
  kmsKeyId: keyvalue 3
provisioner: ebs.csi.aws.com
reclaimPolicy: Delete
volumeBindingMode: WaitForFirstConsumer
EOF
```

1. Indique le nom de la classe de stockage.
2. Le système de fichiers créé sur les volumes provisionnés.
3. Indique le nom de ressource Amazon (ARN) complet de la clé à utiliser lors du chiffrement du volume persistant du conteneur. Lorsque vous ne fournissez pas de clé, mais que le champ crypté est défini sur true, la clé KMS par défaut est utilisée. Consultez la recherche de l'identifiant clé et de la clé ARN sur AWS dans la documentation AWS.

2. Créez une revendication de volume persistante (PVC) avec la classe de stockage spécifiant la clé KMS:

```
$ cat << EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  storageClassName: <storage-class-name>
  resources:
    requests:
      storage: 1Gi
EOF
```

3. Créer des conteneurs de charge de travail pour consommer le PVC:

```
$ cat << EOF | oc create -f -
kind: Pod
```

```
metadata:
  name: mypod
spec:
  containers:
    - name: httpd
      image: quay.io/centos7/httpd-24-centos7
      ports:
        - containerPort: 80
      volumeMounts:
        - mountPath: /mnt/storage
          name: data
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: mypvc
EOF
```

4.1.6. Ressources supplémentaires

- Consultez AWS Elastic Block Store CSI Driver Operator pour obtenir des informations sur l'accès à des options de stockage supplémentaires, telles que des instantanés de volume, qui ne sont pas possibles avec les plugins de volume dans l'arbre.

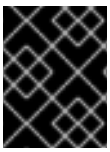
4.2. STOCKAGE PERSISTANT À L'AIDE DU DISQUE PERSISTANT GCE

Le logiciel OpenShift Dedicated prend en charge les volumes de disque persistant GCE (gcePD). Il est possible d'approvisionner votre cluster OpenShift dédié avec un stockage persistant à l'aide de GCE. Certaines familiarités avec Kubernetes et GCE sont supposées.

Le cadre de volume persistant de Kubernetes permet aux administrateurs de fournir un cluster avec un stockage persistant et donne aux utilisateurs un moyen de demander ces ressources sans avoir aucune connaissance de l'infrastructure sous-jacente.

Les volumes de disque persistant GCE peuvent être fournis de manière dynamique.

Les volumes persistants ne sont pas liés à un seul projet ou espace de noms; ils peuvent être partagés sur le cluster OpenShift dédié. Les revendications de volume persistantes sont spécifiques à un projet ou à un espace de noms et peuvent être demandées par les utilisateurs.



IMPORTANT

La disponibilité élevée de stockage dans l'infrastructure est laissée au fournisseur de stockage sous-jacent.

Ressources supplémentaires

- Disque persistant GCE

4.2.1. Création de la classe de stockage GCE

Les classes de stockage sont utilisées pour différencier et délimiter les niveaux et les usages de stockage. En définissant une classe de stockage, les utilisateurs peuvent obtenir des volumes persistants provisionnés dynamiquement.

4.2.2. Création de la revendication de volume persistante

Conditions préalables

Le stockage doit exister dans l'infrastructure sous-jacente avant de pouvoir être monté en tant que volume dans OpenShift Dedicated.

Procédure

1. Dans la console dédiée OpenShift, cliquez sur Stockage → Réclamations de volume persistant.
2. Dans l'aperçu des revendications de volume persistant, cliquez sur Créer une revendication de volume persistant.
3. Définissez les options souhaitées sur la page qui apparaît.
    a. Choisissez la classe de stockage précédemment créée dans le menu déroulant.
    b. Entrez un nom unique pour la revendication de stockage.
    c. Choisissez le mode d'accès. Cette sélection détermine l'accès en lecture et en écriture pour la revendication de stockage.
    d. Définir la taille de la revendication de stockage.
4. Cliquez sur Créer pour créer la revendication de volume persistante et générer un volume persistant.

4.2.3. Format du volume

Avant que OpenShift Dedicated ne monte le volume et le transmet à un conteneur, il vérifie que le volume contient un système de fichiers tel que spécifié par le paramètre fsType dans la définition de volume persistante. Lorsque l'appareil n'est pas formaté avec le système de fichiers, toutes les données de l'appareil sont effacées et l'appareil est automatiquement formaté avec le système de fichiers donné.

Cette vérification vous permet d'utiliser des volumes GCE non formatés comme volumes persistants, car OpenShift Dedicated les formate avant la première utilisation.

CHAPITRE 5. INTERFACE DE STOCKAGE DE CONTENEURS (CSI)

5.1. CONFIGURATION DES VOLUMES CSI

L'interface de stockage de conteneurs (CSI) permet à OpenShift Dedicated de consommer le stockage à partir des extrémités arrière du stockage qui implémentent l'interface CSI en tant que stockage persistant.

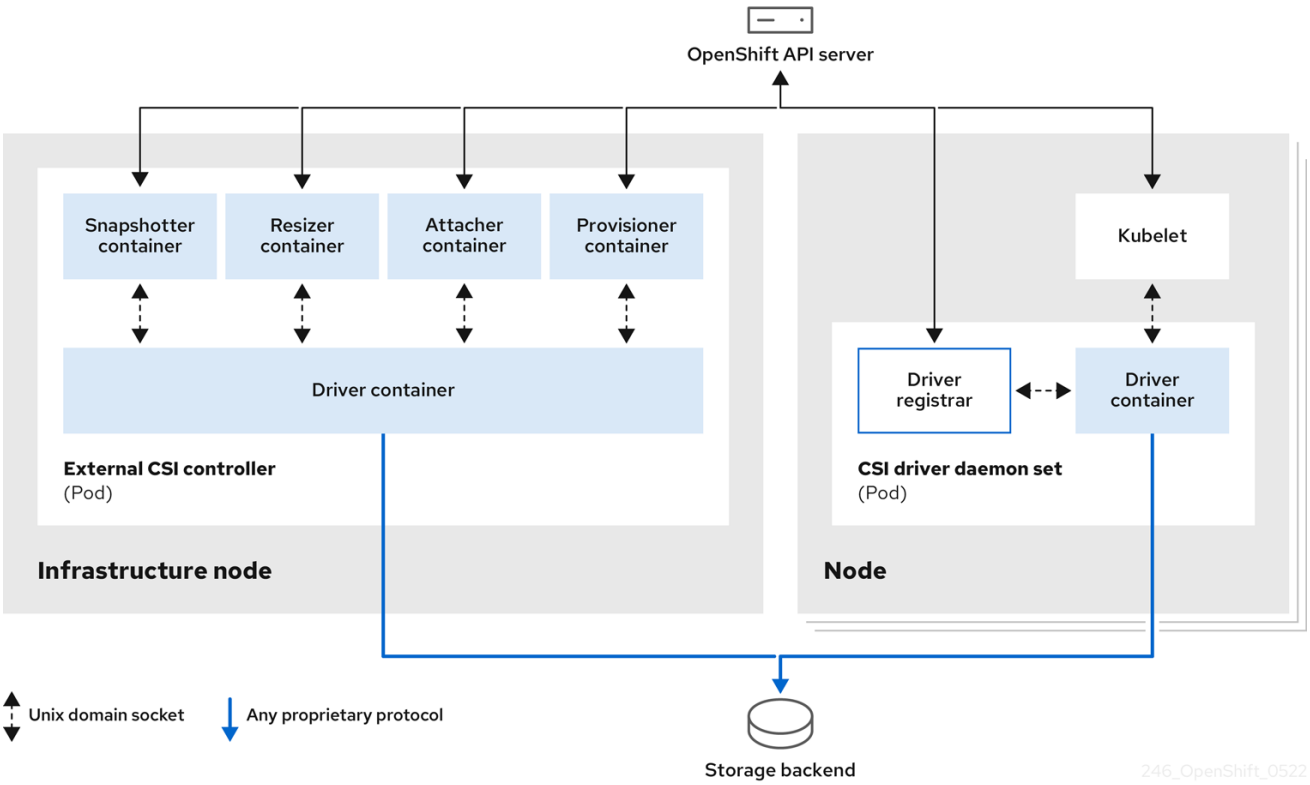
NOTE

Le logiciel OpenShift Dedicated 4 prend en charge la version 1.6.0 de la spécification CSI.

5.1.1. Architecture CSI

Les pilotes CSI sont généralement expédiés sous forme d'images de conteneur. Ces conteneurs ne sont pas au courant d'OpenShift Dedicated où ils s'exécutent. Afin d'utiliser l'arrière de stockage compatible CSI dans OpenShift Dedicated, l'administrateur du cluster doit déployer plusieurs composants qui servent de pont entre OpenShift Dedicated et le pilote de stockage.

Le diagramme suivant fournit une vue d'ensemble de haut niveau sur les composants fonctionnant en pods dans le cluster OpenShift dédié.



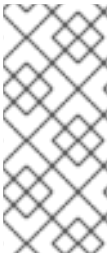
Il est possible d'exécuter plusieurs pilotes CSI pour différentes extrémités de stockage. Chaque pilote a besoin de son propre déploiement de contrôleurs externes et d'un démon défini avec le pilote et le registraire CSI.

5.1.1.1. Contrôleurs CSI externes

Contrôleurs CSI externes est un déploiement qui déploie un ou plusieurs pods avec cinq conteneurs:

- Le conteneur snapshotter montre les objets VolumeSnapshot et VolumeSnapshotContent et est responsable de la création et de la suppression de l'objet VolumeSnapshotContent.
- Le conteneur de resizer est un conteneur sidecar qui surveille les mises à jour et déclenche les opérations ControllerExpandVolume par rapport à un point de terminaison CSI si vous demandez plus de stockage sur l'objet PersistentVolumeClaim.
- Le conteneur de fixation CSI externe traduit les appels d'attache et de détachement d'OpenShift dédiés aux appels respectifs ControllerPublish et ControllerUnpublish au pilote CSI.
- Conteneur de provisionneur CSI externe qui traduit la disposition et supprime les appels d'OpenShift dédiés à leurs appels CreateVolume et DeleteVolume au pilote CSI.
- C) Un conteneur de conducteur CSI.

L'attacheur CSI et les conteneurs de provisionner CSI communiquent avec le conteneur conducteur CSI en utilisant UNIX Domain Sockets, en veillant à ce qu'aucune communication CSI ne quitte la pod. Le conducteur CSI n'est pas accessible de l'extérieur de la gousse.



NOTE

Les opérations de fixation, de détachement, de provision et de suppression exigent généralement que le pilote CSI utilise des informations d'identification sur le backend de stockage. Exécutez les pods de contrôleur CSI sur les nœuds d'infrastructure afin que les informations d'identification ne soient jamais divulguées aux processus utilisateurs, même en cas de violation de sécurité catastrophique sur un nœud de calcul.

NOTE

L'attacheur externe doit également fonctionner pour les pilotes CSI qui ne prennent pas en charge les opérations d'attache ou de détachement de tiers. L'attacheur externe n'émettra aucune opération ControllerPublish ou ControllerUnpublish au pilote CSI. Cependant, il doit toujours s'exécuter pour implémenter l'API de pièce jointe OpenShift Dédicated nécessaire.

5.1.1.2. Jeu de démon de pilote CSI

Le jeu de démon de pilote CSI exécute un pod sur chaque nœud qui permet à OpenShift Dedicated de monter le stockage fourni par le pilote CSI sur le nœud et de l'utiliser dans les charges de travail des utilisateurs (pods) comme volumes persistants (PV). Le pod avec le pilote CSI installé contient les conteneurs suivants:

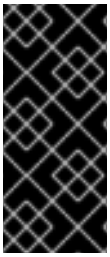
- CSI Driver registrar, qui enregistre le pilote CSI dans le service de nœud ouvert fonctionnant sur le nœud. Le processus openshift-node s'exécute sur le nœud puis se connecte directement au pilote CSI à l'aide du Socket de Domaine UNIX disponible sur le nœud.
- C'est un conducteur de CSI.

Le pilote CSI déployé sur le nœud doit avoir le moins d'informations d'identification à l'arrière du stockage que possible. Le plugin OpenShift Dedicated n'utilisera l'ensemble de plugins de nœud des appels CSI tels que NodePublish/NodeUnpublish et NodeStage/NodeUnstage, si ces appels sont implémentés.

5.1.2. Pilotes CSI pris en charge par OpenShift Dedicated

Le logiciel OpenShift Dedicated installe certains pilotes CSI par défaut, offrant aux utilisateurs des options de stockage qui ne sont pas possibles avec les plugins de volume dans l'arbre.

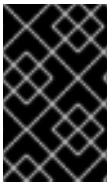
Afin de créer des volumes persistants fournis par CSI qui s'adaptent à ces actifs de stockage pris en charge, OpenShift Dedicated installe par défaut l'opérateur de pilotes CSI nécessaire, le pilote CSI et la classe de stockage requise. Consultez la documentation de l'opérateur CSI pour plus de détails sur l'espace de noms par défaut de l'opérateur et du conducteur par défaut.



IMPORTANT

Les pilotes AWS EFS et GCP Filestore CSI ne sont pas installés par défaut et doivent être installés manuellement. Afin d'obtenir des instructions sur l'installation du pilote AWS EFS CSI, consultez Configuration up AWS Elastic File Service CSI Driver Operator. Afin d'obtenir des instructions sur l'installation du pilote GCP Filestore CSI, consultez Google Compute Platform Filestore CSI Driver Operator.

Le tableau suivant décrit les pilotes CSI qui sont pris en charge par OpenShift Dedicated et les fonctionnalités de CSI qu'ils prennent en charge, tels que les instantanés de volume et le redimensionnement.



IMPORTANT

Dans le cas où votre pilote CSI n'est pas listé dans le tableau suivant, vous devez suivre les instructions d'installation fournies par votre fournisseur de stockage CSI pour utiliser les fonctionnalités CSI prises en charge.

Tableau 5.1. Pilotes et fonctionnalités CSI pris en charge dans OpenShift Dedicated

Conducteur CSI	Instantanés de volume CSI	Aperçus du groupe de volume CSI [1]	Clonage CSI	CSI redimensionnement	Les volumes éphémères en ligne
AWS EBS	▮			▮	
AWS EFS					
Google Compute Platform (GCP) disque persistant (PD)	▮		▮[2]	▮	
GCP Filestore	▮			▮	
Le stockage LVM	▮		▮	▮	

5.1.3. Approvisionnement dynamique

Le provisionnement dynamique du stockage persistant dépend des capacités du pilote CSI et de l'arrière de stockage sous-jacent. Le fournisseur du pilote CSI doit documenter comment créer une classe de stockage dans OpenShift Dedicated et les paramètres disponibles pour la configuration.

La classe de stockage créée peut être configurée pour activer le provisionnement dynamique.

Procédure

- Créez une classe de stockage par défaut qui garantit que tous les PVC qui ne nécessitent aucune classe de stockage spéciale sont fournis par le pilote CSI installé.

```
# oc create -f - << EOF
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <storage-class> 1
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: <provisioner-name> 2
parameters:
EOF
```

1. Le nom de la classe de stockage qui sera créée.
2. Le nom du pilote CSI installé.

5.1.4. Exemple en utilisant le pilote CSI

L'exemple suivant installe un modèle MySQL par défaut sans aucune modification du modèle.

Conditions préalables

- Le pilote CSI a été déployé.
- La classe de stockage a été créée pour le provisionnement dynamique.

Procédure

- Créer le modèle MySQL:

```
# oc new-app mysql-persistent
```

Exemple de sortie

```
--> Deploying template "openshift/mysql-persistent" to project default
...
```

```
# oc get pvc
```

Exemple de sortie

```
NAME          STATUS    VOLUME                                     CAPACITY
```

```
ACCESS MODES   STORAGECLASS   AGE
mysql          Bound     kubernetes-dynamic-pv-3271ffcb4e1811e8   1Gi
RWO            cinder        3s
```

5.2. GESTION DE LA CLASSE DE STOCKAGE PAR DÉFAUT

5.2.1. Aperçu général

La gestion de la classe de stockage par défaut vous permet d'atteindre plusieurs objectifs différents:

- L'application d'un provisionnement statique en désactivant le provisionnement dynamique.
- Lorsque vous avez d'autres classes de stockage préférées, empêchant l'opérateur de stockage de recréer la classe de stockage initiale par défaut.
- En renommant, ou en changeant autrement, la classe de stockage par défaut

Afin d'atteindre ces objectifs, vous modifiez le paramètre du champ spec.storageClassState dans l'objet ClusterCSIDriver. Les paramètres possibles pour ce champ sont:

- L'opérateur de l'interface de stockage de conteneurs (CSI) gère activement sa classe de stockage par défaut, de sorte que la plupart des modifications manuelles apportées par un administrateur de cluster à la classe de stockage par défaut soient supprimées et que la classe de stockage par défaut soit continuellement recréée si vous essayez de la supprimer manuellement.
- Non géré : Vous pouvez modifier la classe de stockage par défaut. L'opérateur CSI ne gère pas activement les classes de stockage, de sorte qu'il ne concilie pas la classe de stockage par défaut qu'il crée automatiquement.
- Les opérateurs CSI suppriment la classe de stockage par défaut.

5.2.2. Gérer la classe de stockage par défaut à l'aide de la console Web

Conditions préalables

- Accès à la console Web dédiée OpenShift.
- Accès au cluster avec des privilèges cluster-admin.

Procédure

Gérer la classe de stockage par défaut à l'aide de la console web:

1. Connectez-vous à la console web.
2. Cliquez sur Administration > CustomResourceDefinitions.
3. Dans la page CustomResourceDefinitions, tapez clustercsidriver pour trouver l'objet ClusterCSIDriver.
4. Cliquez sur ClusterCSIDriver, puis cliquez sur l'onglet Instances.
5. Cliquez sur le nom de l'instance souhaitée, puis cliquez sur l'onglet YAML.

6. Ajoutez le champ spec.storageClassState avec une valeur de Managed, Ungaged ou Removed.

   Exemple :

```
...
spec:
  driverConfig:
    driverType: ''
  logLevel: Normal
  managementState: Managed
  observedConfig: null
  operatorLogLevel: Normal
  storageClassState: Unmanaged (1)
...
```

   (1) champ Sp.storageClassState réglé sur "Non géré"

7. Cliquez sur **Save**.

5.2.3. Gestion de la classe de stockage par défaut à l'aide du CLI

Conditions préalables

- Accès au cluster avec des privilèges cluster-admin.

Procédure

Afin de gérer la classe de stockage à l'aide du CLI, exécutez la commande suivante:

```
oc patch clustercsidriver $DRIVERNAME --type=merge -p "{\"spec\":
{\"storageClassState\":\"${STATE}\"}}" (1)
```

(1) Lorsque ${STATE} est "Supprimé" ou "Géré" ou "Non géré".

Lorsque $DRIVERNAME est le nom du fournisseur. Le nom du provisionneur peut être trouvé en exécutant la commande oc get sc.

5.2.4. Classes de stockage absentes ou multiples par défaut

5.2.4.1. Classes de stockage par défaut multiples

De multiples classes de stockage par défaut peuvent se produire si vous marquez une classe de stockage non par défaut par défaut et ne désglez pas la classe de stockage par défaut existante, ou si vous créez une classe de stockage par défaut lorsqu'une classe de stockage par défaut est déjà présente. Avec plusieurs classes de stockage par défaut présentes, toute revendication de volume persistant (PVC) demandant la classe de stockage par défaut (pvc.spec.storageClassName=nil) obtient la classe de stockage par défaut la plus récente, quel que soit l'état par défaut de cette classe de stockage, et l'administrateur reçoit une alerte dans le tableau de bord des alertes qu'il existe plusieurs classes de stockage par défaut, MultipleDefaultStorageClasses.

5.2.4.2. Absence de classe de stockage par défaut

Il existe deux scénarios possibles où les PVC peuvent tenter d'utiliser une classe de stockage par défaut inexistante:

- L'administrateur supprime la classe de stockage par défaut ou la marque comme non par défaut, puis un utilisateur crée un PVC demandant la classe de stockage par défaut.
- Lors de l'installation, l'installateur crée un PVC demandant la classe de stockage par défaut, qui n'a pas encore été créé.

Dans les scénarios précédents, les PVC restent indéfiniment dans l'état en attente. Afin de résoudre cette situation, créez une classe de stockage par défaut ou déclarez l'une des classes de stockage existantes par défaut. Dès que la classe de stockage par défaut est créée ou déclarée, les PVC obtiennent la nouvelle classe de stockage par défaut. Dans la mesure du possible, les PVC se lient éventuellement à des PV pourvus statiquement ou dynamiquement comme d'habitude, et sortent de l'état en attente.

5.2.5. Changer la classe de stockage par défaut

La procédure suivante permet de modifier la classe de stockage par défaut.

À titre d'exemple, si vous avez deux classes de stockage définies, gp3 et standard, et que vous souhaitez changer la classe de stockage par défaut de gp3 à standard.

Conditions préalables

- Accès au cluster avec des privilèges cluster-admin.

Procédure

Changer la classe de stockage par défaut:

1. Liste des classes de stockage:

```
$ oc get storageclass
```

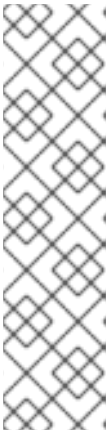
Exemple de sortie

```
NAME            TYPE
gp3 (default)    kubernetes.io/aws-ebs (1)
standard        kubernetes.io/aws-ebs
```

(1) (par défaut) indique la classe de stockage par défaut.

2. Faites de la classe de stockage souhaitée la valeur par défaut.
Dans la classe de stockage souhaitée, définissez l'annotation de classe storageclass.kubernetes.io/is-default-class en exécutant la commande suivante:

```
$ oc patch storageclass standard -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class": "true"}}}'
```



NOTE

Il est possible d'avoir plusieurs classes de stockage par défaut pendant une courte période. Cependant, vous devez vous assurer qu'une seule classe de stockage par défaut existe éventuellement.

Avec plusieurs classes de stockage par défaut présentes, toute revendication de volume persistant (PVC) demandant la classe de stockage par défaut (pvc.spec.storageClassName=nil) obtient la classe de stockage par défaut la plus récente, quel que soit l'état par défaut de cette classe de stockage, et l'administrateur reçoit une alerte dans le tableau de bord des alertes qu'il existe plusieurs classes de stockage par défaut, MultipleDefaultStorageClasses.

3. Supprimez le paramètre de classe de stockage par défaut de l'ancienne classe de stockage par défaut.
   Dans le cas de l'ancienne classe de stockage par défaut, modifiez la valeur de l'annotation de la classe storageclass.kubernetes.io/is-default-class en exécutant la commande suivante:

```
$ oc patch storageclass gp3 -p '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class": "false"}}}'
```

4. Vérifier les changements:

```
$ oc get storageclass
```

Exemple de sortie

```
NAME                 TYPE
gp3                  kubernetes.io/aws-ebs
standard (default)   kubernetes.io/aws-ebs
```

5.3. AWS ELASTIC BLOCK STORE CSI CHAUFFEUR DE PILOTES

5.3.1. Aperçu général

Le logiciel OpenShift Dedicated est capable de fournir des volumes persistants (PV) à l'aide du pilote AWS EBS CSI.

La familiarité avec le stockage persistant et la configuration des volumes CSI est recommandée lorsque vous travaillez avec un opérateur et un pilote d'interface de stockage de conteneurs (CSI).

Afin de créer des PV fournis par CSI qui sont montés sur les ressources de stockage AWS EBS, OpenShift Dedicated installe par défaut l'opérateur de pilotes AWS EBS CSI (un opérateur Red Hat) et le pilote AWS EBS CSI par défaut dans l'espace de noms openshift-cluster-csi-drivers.

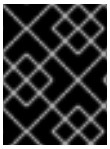
- L'opérateur de pilotes AWS EBS CSI fournit une classe de stockage par défaut que vous pouvez utiliser pour créer des PVC. Le cas échéant, vous pouvez désactiver cette classe de stockage par défaut (voir Gestion de la classe de stockage par défaut). En outre, vous avez la possibilité de créer AWS EBS StorageClass comme décrit dans Stockage persistant à l'aide d'Amazon Elastic Block Store.

- Le pilote AWS EBS CSI vous permet de créer et de monter des PV AWS EBS.

5.3.2. À propos de CSI

Les fournisseurs de stockage ont traditionnellement fourni des pilotes de stockage dans le cadre de Kubernetes. Avec l'implémentation de l'interface de stockage de conteneurs (CSI), les fournisseurs tiers peuvent fournir des plugins de stockage à l'aide d'une interface standard sans jamais avoir à modifier le code Kubernetes de base.

Les opérateurs CSI offrent aux utilisateurs d'OpenShift des options de stockage, telles que des instantanés de volume, qui ne sont pas possibles avec les plugins de volume dans l'arbre.



IMPORTANT

Dédié par défaut à l'utilisation du plugin CSI pour fournir le stockage Amazon Elastic Block Store (Amazon EBS).

Afin d'obtenir des informations sur le provisionnement dynamique des volumes persistants AWS EBS dans OpenShift Dedicated, consultez le stockage persistant à l'aide d'Amazon Elastic Block Store.

Ressources supplémentaires

- Le stockage persistant à l'aide d'Amazon Elastic Block Store
- Configuration des volumes CSI

5.4. AWS ELASTIC FILE SERVICE CSI DRIVER OPERATOR

IMPORTANT

Cette procédure est spécifique à l'opérateur de conducteur AWS EFS CSI (un opérateur Red Hat), qui n'est applicable qu'aux versions 4.10 et ultérieures d'OpenShift Dedicated.

5.4.1. Aperçu général

Le logiciel OpenShift Dedicated est capable de fournir des volumes persistants (PV) à l'aide du pilote CSI (Container Storage Interface) pour AWS Elastic File Service (EFS).

La familiarité avec le stockage persistant et la configuration des volumes CSI est recommandée lorsque vous travaillez avec un opérateur CSI et un conducteur.

Après avoir installé l'opérateur de pilotes AWS EFS CSI, OpenShift Dedicated installe par défaut le pilote AWS EFS CSI Operator et AWS EFS CSI dans l'espace de noms openshift-cluster-csi-drivers. Cela permet à AWS EFS CSI Driver Operator de créer des PV fournis par CSI qui s'adaptent aux actifs AWS EFS.

- L'opérateur de pilotes AWS EFS CSI, après avoir été installé, ne crée pas de classe de stockage par défaut à utiliser pour créer des revendications de volume persistantes (PVC). Cependant, vous pouvez créer manuellement la classe de stockage AWS EFS. AWS EFS CSI Driver Operator prend en charge le provisionnement dynamique de volume en permettant la création de volumes de stockage à la demande. Cela élimine la nécessité pour les administrateurs de clusters de pré-fournir le stockage.
- Le pilote AWS EFS CSI vous permet de créer et de monter des PV AWS EFS.

NOTE

AWS EFS ne prend en charge que les volumes régionaux, et non les volumes zonaux.

5.4.2. À propos de CSI

Les fournisseurs de stockage ont traditionnellement fourni des pilotes de stockage dans le cadre de Kubernetes. Avec l'implémentation de l'interface de stockage de conteneurs (CSI), les fournisseurs tiers peuvent fournir des plugins de stockage à l'aide d'une interface standard sans jamais avoir à modifier le code Kubernetes de base.

Les opérateurs CSI offrent aux utilisateurs d'OpenShift des options de stockage, telles que des instantanés de volume, qui ne sont pas possibles avec les plugins de volume dans l'arbre.

5.4.3. Configuration de l'opérateur de pilotes AWS EFS CSI

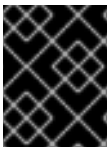
1. Lorsque vous utilisez AWS EFS avec AWS Secure Token Service (STS), obtenez un rôle Amazon Resource Name (ARN) pour STS. Ceci est nécessaire pour installer l'opérateur de pilotes AWS EFS CSI.

2. Installez l'opérateur de pilotes AWS EFS CSI.

3. Installez le pilote AWS EFS CSI.

5.4.3.1. L'obtention d'un rôle Amazon Resource Name for Security Token Service

Cette procédure explique comment obtenir un rôle Amazon Resource Name (ARN) pour configurer l'opérateur de pilotes AWS EFS CSI avec OpenShift dédié sur AWS Security Token Service (STS).



IMPORTANT

Effectuez cette procédure avant d'installer AWS EFS CSI Driver Operator (voir Installation de la procédure AWS EFS CSI Driver Operator).

Conditions préalables

- Accès au cluster en tant qu'utilisateur avec le rôle cluster-admin.
- Identifiants de compte AWS

Procédure

1. Créez un fichier JSON de stratégie IAM avec le contenu suivant:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeAccessPoints",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeMountTargets",
        "ec2:DescribeAvailabilityZones",
        "elasticfilesystem:TagResource"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateAccessPoint"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:RequestTag/efs.csi.aws.com/cluster": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:DeleteAccessPoint",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/efs.csi.aws.com/cluster": "true"
        }
      }
    }
  ]
}
```

2. Créez un fichier JSON de confiance IAM avec le contenu suivant:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<your_aws_account_ID>:oidc-
provider/<openshift_oidc_provider>" (1)
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<openshift_oidc_provider>:sub": [ (2)
            "system:serviceaccount:openshift-cluster-csi-drivers:aws-efs-csi-driver-operator",
            "system:serviceaccount:openshift-cluster-csi-drivers:aws-efs-csi-driver-controller-sa"
          ]
        }
      }
    }
  ]
}
```

(1) Indiquez votre identifiant de compte AWS et le point de terminaison du fournisseur OpenShift OIDC.

Achetez votre identifiant de compte AWS en exécutant la commande suivante:

```
$ aws sts get-caller-identity --query Account --output text
```

Obtenez le point de terminaison OpenShift OIDC en exécutant la commande suivante:

```
$ openshift_oidc_provider=`oc get authentication.config.openshift.io cluster \
  -o json | jq -r .spec.serviceAccountIssuer | sed -e "s/^https:\/\///"`; \
  echo $openshift_oidc_provider
```

2 Indiquez à nouveau le point de terminaison OpenShift OIDC.

3. Créer le rôle de l'IAM:

```
ROLE_ARN=$(aws iam create-role \
  --role-name "<your_cluster_name>-aws-efs-csi-operator" \
  --assume-role-policy-document file://<your_trust_file_name>.json \
  --query "Role.Arn" --output text); echo $ROLE_ARN
```

Copiez le rôle ARN. Lors de l'installation de l'opérateur de pilotes AWS EFS CSI, vous en aurez besoin.

4. Créer la politique IAM:

```
POLICY_ARN=$(aws iam create-policy \
  --policy-name "<your_cluster_name>-aws-efs-csi" \
  --policy-document file://<your_policy_file_name>.json \
  --query 'Policy.Arn' --output text); echo $POLICY_ARN
```

5. Attacher la politique de l'IAM au rôle de l'IAM:

```
$ aws iam attach-role-policy \
  --role-name "<your_cluster_name>-aws-efs-csi-operator" \
  --policy-arn $POLICY_ARN
```

Les prochaines étapes

Installez l'opérateur de pilotes AWS EFS CSI.

Ressources supplémentaires

- Installation de l'opérateur de pilotes AWS EFS CSI
- Installation du pilote AWS EFS CSI

5.4.3.2. Installation de l'opérateur de pilotes AWS EFS CSI

L'opérateur de pilotes AWS EFS CSI (un opérateur Red Hat) n'est pas installé dans OpenShift Dédié par défaut. Installez et configurez l'opérateur de pilotes AWS EFS CSI dans votre cluster.

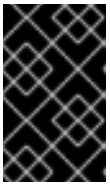
Conditions préalables

- Accès à la console Web dédiée OpenShift.

Procédure

Installer l'opérateur de pilotes AWS EFS CSI à partir de la console web:

1. Connectez-vous à la console web.
2. Installez l'opérateur AWS EFS CSI:
   a. Cliquez sur Opérateurs → OperatorHub.
   b. Localisez l'opérateur AWS EFS CSI en tapant AWS EFS CSI dans la zone de filtre.
   c. Cliquez sur le bouton AWS EFS CSI Driver Operator.



IMPORTANT

Assurez-vous de sélectionner l'opérateur de conducteur AWS EFS CSI et non l'opérateur EFS AWS. L'opérateur AWS EFS est un opérateur communautaire et n'est pas pris en charge par Red Hat.

   a. Dans la page AWS EFS CSI Driver Operator, cliquez sur Installer.
   b. Dans la page Installer l'opérateur, assurez-vous que:
      - Lorsque vous utilisez AWS EFS avec AWS Secure Token Service (STS), dans le champ ARN du rôle, entrez le rôle ARN copié à partir de la dernière étape de la procédure d'obtention d'un rôle Amazon Resource Name for Security Token Service.
      - L'ensemble des espaces de noms du cluster (par défaut) est sélectionné.
      - Installé Namespace est réglé sur openshift-cluster-csi-drivers.
   c. Cliquez sur **Install**.
   Après la fin de l'installation, l'opérateur AWS EFS CSI est listé dans la section Opérateurs installés de la console Web.

Les prochaines étapes

Installez le pilote AWS EFS CSI.

5.4.3.3. Installation du pilote AWS EFS CSI

Après avoir installé le pilote AWS EFS CSI Driver Operator (un opérateur Red Hat), vous installez le pilote AWS EFS CSI.

Conditions préalables

- Accès à la console Web dédiée OpenShift.

Procédure

1. Cliquez sur Administration → CustomResourceDefinitions → ClusterCSIDriver.
2. Dans l'onglet Instances, cliquez sur Créer un clusterCSIDriver.
3. Consultez le fichier YAML suivant:

```
apiVersion: operator.openshift.io/v1
kind: ClusterCSIDriver
metadata:
    name: efs.csi.aws.com
spec:
  managementState: Managed
```

4. Cliquez sur **Create**.
5. Attendez que les Conditions suivantes changent pour un statut « Vrai »:
   - AWSEFSDriverNodeServiceControllerAvailable
   - AWSEFSDriverControllerServiceControllerDisponible

5.4.4. Création de la classe de stockage AWS EFS

Les classes de stockage sont utilisées pour différencier et délimiter les niveaux et les usages de stockage. En définissant une classe de stockage, les utilisateurs peuvent obtenir des volumes persistants provisionnés dynamiquement.

L'opérateur de pilotes AWS EFS CSI (un opérateur Red Hat), après avoir été installé, ne crée pas de classe de stockage par défaut. Cependant, vous pouvez créer manuellement la classe de stockage AWS EFS.

5.4.4.1. Création de la classe de stockage AWS EFS à l'aide de la console

Procédure

1. Dans la console dédiée OpenShift, cliquez sur Storage → StorageClasses.
2. Dans la page StorageClasses, cliquez sur Créer une classe de stockage.
3. Dans la page StorageClass, effectuez les étapes suivantes:
   a. Entrez un nom pour référencer la classe de stockage.
   b. Facultatif: Entrez la description.
   c. Choisissez la politique de récupération.
   d. Choisissez efs.csi.aws.com dans la liste déroulante Provisioner.
   e. Facultatif : Définir les paramètres de configuration du provisionneur sélectionné.
4. Cliquez sur **Create**.

5.4.4.2. Création de la classe de stockage AWS EFS à l'aide du CLI

Procédure

- Créer un objet StorageClass:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
```

```
metadata:
  name: efs-sc
provisioner: efs.csi.aws.com
parameters:
  provisioningMode: efs-ap 1
  fileSystemId: fs-a5324911 2
  directoryPerms: "700" 3
  gidRangeStart: "1000" 4
  gidRangeEnd: "2000" 5
  basePath: "/dynamic_provisioning" 6
```

1. provisionnementMode doit être efs-ap pour permettre un provisionnement dynamique.
2. fileSystemId doit être l'ID du volume EFS créé manuellement.
3. directoryPerms est l'autorisation par défaut du répertoire racine du volume. Dans cet exemple, le volume n'est accessible que par le propriétaire.
4. 5 gidRangeStart et gidRangeEnd définissent la plage des ID de groupe POSIX (GID) qui sont utilisés pour définir le GID du point d'accès AWS. Dans le cas contraire, la plage par défaut est 50000-7000000. Chaque volume provisionné, et donc le point d'accès AWS, se voit attribuer un GID unique de cette plage.
6. basePath est le répertoire sur le volume EFS qui est utilisé pour créer des volumes provisionnés dynamiquement. Dans ce cas, un PV est fourni comme "/dynamic_provisioning/<random uuid>" sur le volume EFS. Le sous-répertoire n'est monté qu'à des pods utilisant le PV.

NOTE

L'administrateur de cluster peut créer plusieurs objets StorageClass, chacun utilisant un volume EFS différent.

5.4.5. Création et configuration de l'accès aux volumes EFS dans AWS

Cette procédure explique comment créer et configurer des volumes EFS dans AWS afin que vous puissiez les utiliser dans OpenShift Dedicated.

Conditions préalables

- Identifiants de compte AWS

Procédure

Créer et configurer l'accès à un volume EFS dans AWS:

1. Dans la console AWS, ouvrez https://console.aws.amazon.com/efs.
2. Cliquez sur Créer un système de fichiers:
   - Entrez un nom pour le système de fichiers.
   - Dans le Cloud privé virtuel (VPC), sélectionnez le cloud privé virtuel (VPC) de votre OpenShift Dedicated.

- Acceptez les paramètres par défaut pour toutes les autres sélections.

3. Attendez que le volume et monter les cibles pour terminer soient entièrement créés:

   a. Allez à https://console.aws.amazon.com/efs#/file-systems.

   b. Cliquez sur votre volume, et dans l'onglet Réseau attendez que toutes les cibles de montage deviennent disponibles (~1-2 minutes).

4. Dans l'onglet Réseau, copiez l'ID du groupe de sécurité (vous en aurez besoin à l'étape suivante).

5. Allez à https://console.aws.amazon.com/ec2/v2/home#SecurityGroups, et trouvez le groupe de sécurité utilisé par le volume EFS.

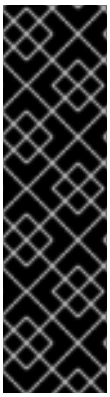
6. Dans l'onglet règles entrantes, cliquez sur Modifier les règles entrantes, puis ajoutez une nouvelle règle avec les paramètres suivants pour permettre à OpenShift Dedicated nodes d'accéder aux volumes EFS:

   - Catégorie: NFS
   - Le protocole: TCP
   - Gamme de ports: 2049
   - Gamme d'adresses personnalisées/IP de vos nœuds (par exemple: "10.0.0.0/16")
     Cette étape permet à OpenShift Dedicated d'utiliser les ports NFS à partir du cluster.

7. Gardez la règle.

5.4.6. Provisionnement dynamique pour Amazon Elastic File Storage

Le pilote AWS EFS CSI prend en charge une forme de provisionnement dynamique différente de celle des autres pilotes CSI. Il prévoit les nouvelles PV en tant que sous-répertoires d'un volume EFS préexistant. Les PV sont indépendants les uns des autres. Cependant, ils partagent tous le même volume EFS. Lorsque le volume est supprimé, tous les PV mis à disposition de celui-ci sont également supprimés. Le pilote EFS CSI crée un point d'accès AWS pour chaque sous-répertoire. En raison des limites AWS AccessPoint, vous ne pouvez fournir dynamiquement 1000 PV à partir d'un seul volume StorageClass/EFS.



IMPORTANT

Il est à noter que PVC.spec.resources n'est pas appliqué par EFS.

Dans l'exemple ci-dessous, vous demandez 5 GiB d'espace. Cependant, le PV créé est illimité et peut stocker n'importe quelle quantité de données (comme les pétaoctets). L'application cassée, ou même une application voyou, peut entraîner des dépenses importantes lorsqu'elle stocke trop de données sur le volume.

Il est fortement recommandé d'utiliser la surveillance des tailles de volume EFS dans AWS.

Conditions préalables

- Les volumes Amazon Elastic File Storage (Amazon EFS) ont été créés.

- La classe de stockage AWS EFS a été créée.

Procédure

Afin de permettre un approvisionnement dynamique:

- Créez un PVC (ou StatefulSet ou Template) comme d'habitude, en se référant à la classe de stockage créée précédemment.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test
spec:
  storageClassName: efs-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
```

En cas de difficulté à configurer un provisionnement dynamique, consultez le dépannage AWS EFS.

Ressources supplémentaires

- Création et configuration de l'accès à AWS EFS volume(s)
- Création de la classe de stockage AWS EFS

5.4.7. Créer des PV statiques avec Amazon Elastic File Storage

Il est possible d'utiliser un volume Amazon Elastic File Storage (Amazon EFS) comme un seul PV sans provisionnement dynamique. L'ensemble du volume est monté sur des pods.

Conditions préalables

- Les volumes Amazon EFS ont été créés.

Procédure

- Créez le PV à l'aide du fichier YAML suivant:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: efs-pv
spec:
  capacity: 1
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  csi:
```

```
    driver: efs.csi.aws.com
    volumeHandle: fs-ae66151a 2
    volumeAttributes:
      encryptInTransit: "false" 3
```

1 la spécification.capacity n'a aucun sens et est ignorée par le pilote CSI. Il n'est utilisé que lors de la fixation à un PVC. Les applications peuvent stocker n'importe quelle quantité de données dans le volume.

2 le volumeHandle doit être le même ID que le volume EFS que vous avez créé dans AWS. Lorsque vous fournissez votre propre point d'accès, VolumeHandle doit être <EFS volume ID>::<ID de point d'accès>. Exemple: fs-6e633ada::fsap-081a1d293f0004630.

3 Au besoin, vous pouvez désactiver le cryptage en transit. Le chiffrement est activé par défaut.

En cas de problème de configuration des PV statiques, consultez le dépannage AWS EFS.

5.4.8. Amazon Elastic File Storage Sécurité

Les informations suivantes sont importantes pour la sécurité Amazon Elastic File Storage (Amazon EFS).

Lors de l'utilisation des points d'accès, par exemple, en utilisant le provisionnement dynamique comme décrit précédemment, Amazon remplace automatiquement les GID sur les fichiers par le GID du point d'accès. En outre, EFS considère l'ID utilisateur, l'ID de groupe et les ID de groupe secondaires du point d'accès lors de l'évaluation des autorisations du système de fichiers. EFS ignore les identifiants du client NFS. Consultez https://docs.aws.amazon.com/efs/latest/ug/efs-access-points.html pour plus d'informations sur les points d'accès.

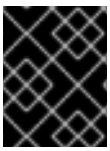
En conséquence, les volumes EFS ignorent silencieusement FSGroup; OpenShift Dedicated n'est pas en mesure de remplacer les GID des fichiers sur le volume par FSGroup. Chaque pod pouvant accéder à un point d'accès EFS monté peut accéder à n'importe quel fichier dessus.

En dehors de cela, le cryptage en transit est activé par défaut. Consultez https://docs.aws.amazon.com/efs/latest/ug/encryption-in-transit.html pour plus d'informations.

5.4.9. AWS EFS stockage CSI métriques d'utilisation

5.4.9.1. Aperçu des métriques d'utilisation

Les métriques d'utilisation de l'interface de stockage de conteneurs (EFS) d'Amazon Web Services (AWS) vous permettent de surveiller la quantité d'espace utilisée par les volumes EFS fournis de manière dynamique ou statique.



IMPORTANT

Ces fonctionnalités sont désactivées par défaut, car l'activation des métriques peut entraîner une dégradation des performances.

La fonction métriques d'utilisation AWS EFS collecte les métriques de volume dans le pilote AWS EFS CSI en parcourant de manière récursive les fichiers dans le volume. Comme cet effort peut dégrader les performances, les administrateurs doivent explicitement activer cette fonctionnalité.

5.4.9.2. Activer les métriques d'utilisation à l'aide de la console Web

Activer les paramètres d'utilisation de l'interface de stockage des conteneurs de stockage (EFS) d'Amazon Web Services (AWS) à l'aide de la console Web:

1. Cliquez sur Administration > CustomResourceDefinitions.
2. Dans la page CustomResourceDefinitions à côté de la zone déroulante Nom, tapez clustercsidriver.
3. Cliquez sur CRD ClusterCSIDriver.
4. Cliquez sur l'onglet YAML.
5. Dans spec.aws.efsVolumeMetrics.state, définissez la valeur à RecursiveWalk.
   La RecursiveWalk indique que la collecte de mesures de volume dans le pilote AWS EFS CSI est effectuée en parcourant de manière récursive les fichiers dans le volume.

   Exemple ClusterCSIDriver efs.csi.aws.com fichier YAML

```
spec:
  driverConfig:
    driverType: AWS
    aws:
      efsVolumeMetrics:
        state: RecursiveWalk
        recursiveWalk:
          refreshPeriodMinutes: 100
          fsRateLimit: 10
```

6. Facultatif: Pour définir comment fonctionne la marche récursive, vous pouvez également définir les champs suivants:
   - rafraîchissementPeriodMinutes: Spécifie la fréquence de rafraîchissement des métriques de volume en quelques minutes. Lorsque ce champ est laissé vide, un défaut raisonnable est choisi, qui est susceptible de changer au fil du temps. La valeur par défaut actuelle est 240 minutes. La plage valide est de 1 à 43 200 minutes.
   - fsRateLimit: Définit la limite de taux pour le traitement des métriques de volume dans les goroutines par système de fichiers. Lorsque ce champ est laissé vide, un défaut raisonnable est choisi, qui est susceptible de changer au fil du temps. La valeur par défaut actuelle est 5 goroutines. La gamme valide est de 1 à 100 goroutines.
7. Cliquez sur **Save**.

NOTE

Afin de désactiver les métriques d'utilisation AWS EFS CSI, utilisez la procédure précédente, mais pour spec.aws.efsVolumeMetrics.state, modifiez la valeur de RecursiveWalk à Disabled.

5.4.9.3. Activer les métriques d'utilisation à l'aide du CLI

Activer les paramètres d'utilisation de l'interface de stockage des conteneurs (EFS) d'Amazon Web Services (AWS) à l'aide du CLI:

1. Editez ClusterCSIDriver en exécutant la commande suivante:

```
$ oc edit clustercsidriver efs.csi.aws.com
```

2. Dans spec.aws.efsVolumeMetrics.state, définissez la valeur à RecursiveWalk.
La RecursiveWalk indique que la collecte de mesures de volume dans le pilote AWS EFS CSI est effectuée en parcourant de manière récursive les fichiers dans le volume.

Exemple ClusterCSIDriver efs.csi.aws.com fichier YAML

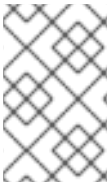
```
spec:
  driverConfig:
    driverType: AWS
    aws:
      efsVolumeMetrics:
        state: RecursiveWalk
        recursiveWalk:
          refreshPeriodMinutes: 100
          fsRateLimit: 10
```

3. Facultatif: Pour définir comment fonctionne la marche récursive, vous pouvez également définir les champs suivants:

   - rafraîchissementPeriodMinutes: Spécifie la fréquence de rafraîchissement des métriques de volume en quelques minutes. Lorsque ce champ est laissé vide, un défaut raisonnable est choisi, qui est susceptible de changer au fil du temps. La valeur par défaut actuelle est 240 minutes. La plage valide est de 1 à 43 200 minutes.

   - fsRateLimit: Définit la limite de taux pour le traitement des métriques de volume dans les goroutines par système de fichiers. Lorsque ce champ est laissé vide, un défaut raisonnable est choisi, qui est susceptible de changer au fil du temps. La valeur par défaut actuelle est 5 goroutines. La gamme valide est de 1 à 100 goroutines.

4. Enregistrez les modifications de l'objet efs.csi.aws.com.



NOTE

Afin de désactiver les métriques d'utilisation AWS EFS CSI, utilisez la procédure précédente, mais pour spec.aws.efsVolumeMetrics.state, modifiez la valeur de RecursiveWalk à Disabled.

5.4.10. Dépannage d'Amazon Elastic File Storage

Les informations suivantes fournissent des conseils sur la façon de résoudre les problèmes avec Amazon Elastic File Storage (Amazon EFS):

- L'opérateur AWS EFS et le pilote CSI fonctionnent dans l'espace de noms openshift-cluster-csi-drivers.

- Afin d'initier la collecte des logs de l'opérateur AWS EFS et du pilote CSI, exécutez la commande suivante:

```
$ oc adm must-gather
[must-gather      ] OUT Using must-gather plugin-in image: quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:125f183d13601537ff15b3239df95d47f0a604da2847b561151fedd699f5e3a5
[must-gather      ] OUT namespace/openshift-must-gather-xm4wq created
[must-gather      ] OUT clusterrolebinding.rbac.authorization.k8s.io/must-gather-2bd8x created
[must-gather      ] OUT pod for plug-in image quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:125f183d13601537ff15b3239df95d47f0a604da2847b561151fedd699f5e3a5 created
```

- Afficher les erreurs AWS EFS Operator, afficher le statut ClusterCSIDriver:

```
$ oc get clustercsidriver efs.csi.aws.com -o yaml
```

- B) Si un volume ne peut pas être monté sur un pod (comme indiqué dans la sortie de la commande suivante):

```
$ oc describe pod
...
  Type     Reason       Age    From               Message
  ----     ------       ----   ----               -------
  Normal   Scheduled    2m13s  default-scheduler  Successfully assigned default/efs-app to ip-10-0-135-94.ec2.internal
  Warning  FailedMount  13s    kubelet            MountVolume.SetUp failed for volume "pvc-d7c097e6-67ec-4fae-b968-7e7056796449" : rpc error: code = DeadlineExceeded desc = context deadline exceeded (1)
  Warning  FailedMount  10s    kubelet            Unable to attach or mount volumes: unmounted volumes=[persistent-storage], unattached volumes=[persistent-storage kube-api-access-9j477]: timed out waiting for the condition
```

(1) Le message d'avertissement indiquant le volume non monté.

Cette erreur est souvent causée par la chute de paquets AWS entre un nœud dédié OpenShift et Amazon EFS.

Assurez-vous que les éléments suivants sont corrects:

  - Groupes de pare-feu et de sécurité AWS
  - La mise en réseau: numéro de port et adresses IP

5.4.11. Désinstallation de l'opérateur de pilotes AWS EFS CSI

Les EFS PV sont inaccessibles après avoir désinstallé AWS EFS CSI Driver Operator (un opérateur Red Hat).

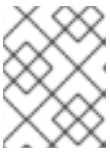
Conditions préalables

- Accès à la console Web dédiée OpenShift.

Procédure

Désinstaller AWS EFS CSI Driver Operator de la console web:

1. Connectez-vous à la console web.
2. Arrêtez toutes les applications utilisant AWS EFS PV.
3. Effacer tous les PV AWS EFS:
   a. Cliquez sur Stockage → PersistentVolumeClaims.
   b. Choisissez chaque PVC utilisé par AWS EFS CSI Driver Operator, cliquez sur le menu déroulant situé à l'extrême droite du PVC, puis cliquez sur Supprimer PersistentVolumeClaims.
4. Désinstaller le pilote AWS EFS CSI:



NOTE

Avant de pouvoir désinstaller l'opérateur, vous devez d'abord supprimer le pilote CSI.

   a. Cliquez sur Administration → CustomResourceDefinitions → ClusterCSIDriver.
   b. Dans l'onglet Instances, pour efs.csi.aws.com, à l'extrême gauche, cliquez sur le menu déroulant, puis cliquez sur Supprimer ClusterCSIDriver.
   c. Lorsque vous l'invitez, cliquez sur Supprimer.
5. Désinstallez l'opérateur AWS EFS CSI:
   a. Cliquez sur Opérateurs → Opérateurs installés.
   b. Dans la page Opérateurs installés, faites défiler ou tapez AWS EFS CSI dans la zone Recherche par nom pour trouver l'opérateur, puis cliquez dessus.
   c. En haut, à droite de la page Opérateurs installés > Détails de l'opérateur, cliquez sur Actions → Désinstaller l'opérateur.
   d. Lorsque vous êtes invité dans la fenêtre de désinstallation de l'opérateur, cliquez sur le bouton Désinstaller pour supprimer l'opérateur de l'espace de noms. Les applications déployées par l'opérateur sur le cluster doivent être nettoyées manuellement.
   Après la désinstallation, AWS EFS CSI Driver Operator n'est plus listé dans la section Opérateurs installés de la console Web.

NOTE

Avant de pouvoir détruire un cluster (openshift-install destroy cluster), vous devez supprimer le volume EFS dans AWS. Aucun cluster dédié OpenShift ne peut être détruit lorsqu'il y a un volume EFS qui utilise le VPC du cluster. Amazon n'autorise pas la suppression d'un tel VPC.

5.4.12. Ressources supplémentaires

- Configuration des volumes CSI

5.5. GESTIONNAIRE DE PILOTES GCP PD CSI

5.5.1. Aperçu général

Le logiciel OpenShift Dedicated peut fournir des volumes persistants (PV) à l'aide du pilote CSI (Container Storage Interface) pour le stockage persistant sur disque (PD) de Google Cloud Platform (GCP).

La familiarité avec le stockage persistant et la configuration des volumes CSI est recommandée lorsque vous travaillez avec un opérateur et un pilote d'interface de stockage de conteneurs (CSI).

Afin de créer des volumes persistants (PV) fournis par CSI qui s'adaptent aux ressources de stockage GCP PD, OpenShift Dedicated installe par défaut l'opérateur de pilotes GCP PD CSI et le pilote GCP PD CSI dans l'espace de noms openshift-cluster-csi-drivers.

- GCP PD CSI Driver Operator: Par défaut, l'opérateur fournit une classe de stockage que vous pouvez utiliser pour créer des PVC. Le cas échéant, vous pouvez désactiver cette classe de stockage par défaut (voir Gestion de la classe de stockage par défaut). Il est également possible de créer la classe de stockage GCP PD décrite dans Stockage persistant à l'aide de GCE Persistent Disk.

- Pilote PD GCP: Le pilote vous permet de créer et de monter des PV GCP PD.
  Le pilote GCP PD CSI prend en charge le type d'instance C3 pour les séries de machines en métal nu et N4. Le type d'instance C3 et la série de machines N4 prennent en charge les disques hyperdisques.

5.5.2. C3 type d'instance pour la série de machines en métal nu et N4

5.5.2.1. Limites de type d'instance C3 et N4

Le support du pilote GCP PD CSI pour le type d'instance C3 pour les séries de machines en métal nu et N4 présente les limites suivantes:

- Les volumes de clonage ne sont pas pris en charge lors de l'utilisation de pools de stockage.

- En cas de clonage ou de redimensionnement, la taille du volume d'origine des disques hyperdisques doit être de 6Gi ou plus.

- La classe de stockage par défaut est standard-csi.

IMPORTANT

Il faut créer manuellement une classe de stockage.

En ce qui concerne la création de la classe de stockage, consultez l'étape 2 de la section Configuration des disques hyperdisques.

- Les clusters avec machines virtuelles mixtes (VM) qui utilisent différents types de stockage, par exemple N2 et N4, ne sont pas pris en charge. Cela est dû au fait que les disques hyperdisques ne sont pas utilisables sur la plupart des machines virtuelles héritées. De même, les disques persistants réguliers ne sont pas utilisables sur les machines virtuelles N4/C3.

- Le cluster GCP avec des nœuds c3-standard-2, c3-standard-4, n4-standard-2 et n4-standard-4 peut dépasser à tort le nombre maximum de disque fixe, ce qui devrait être 16 (lien JIRA).

5.5.2.2. Aperçu des pools de stockage pour disques hyperdisques

Les pools de stockage Hyperdisk peuvent être utilisés avec Compute Engine pour le stockage à grande échelle. Le pool de stockage hyperdisque est une collection achetée de capacité, de débit et d'IOPS, que vous pouvez ensuite prévoir pour vos applications au besoin. Il est possible d'utiliser des pools de stockage hyperdisque pour créer et gérer des disques dans des pools et utiliser les disques sur plusieurs charges de travail. En gérant les disques en agrégat, vous pouvez économiser des coûts tout en réalisant une croissance de la capacité et des performances attendues. En utilisant uniquement le stockage dont vous avez besoin dans les pools de stockage hyperdisque, vous réduisez la complexité de la capacité de prévision et réduisez la gestion en passant de la gestion de centaines de disques à la gestion d'un seul pool de stockage.

Afin de configurer des pools de stockage, consultez Configuration des disques hyperdisque.

5.5.2.3. Configuration de disques hyperdisk-équilibrés

Conditions préalables

- Accès au cluster avec privilèges administratifs

Procédure

Configurer des disques hyperdisk-équilibrés:

1. Créez une revendication de volume persistante (PVC) qui utilise la classe de stockage hyperdisque spécifique à l'aide de l'exemple de fichier YAML suivant:

   Exemple de fichier PVC YAML

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: my-pvc
   spec:
     storageClassName: hyperdisk-sc 1
     accessModes:
     - ReadWriteOnce
     resources:
       requests:
         storage: 2048Gi 2
   ```

   1. Le PVC fait référence à la classe de stockage spécifique à la piscine de stockage. Dans cet exemple, hyperdisk-sc.

   2. Capacité de stockage cible du volume hyperdisque équilibré. Dans cet exemple, 2048Gi.

2. Créez un déploiement qui utilise le PVC que vous venez de créer. L'utilisation d'un déploiement permet de s'assurer que votre application a accès au stockage persistant même après le redémarrage et le rééchelonnement du pod:

   a. Assurez-vous qu'un pool de nœuds avec la série de machines spécifiée est opérationnel avant de créer le déploiement. Dans le cas contraire, la gousse ne parvient pas à planifier.

   b. À l'aide de l'exemple suivant fichier YAML pour créer le déploiement:

Exemple de déploiement du fichier YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres
spec:
  selector:
    matchLabels:
      app: postgres
  template:
    metadata:
      labels:
        app: postgres
    spec:
      nodeSelector:
        cloud.google.com/machine-family: n4 (1)
      containers:
      - name: postgres
        image: postgres:14-alpine
        args: [ "sleep", "3600" ]
        volumeMounts:
        - name: sdk-volume
          mountPath: /usr/share/data/
      volumes:
      - name: sdk-volume
        persistentVolumeClaim:
          claimName: my-pvc (2)
```

(1) Indique la famille de machines. Dans cet exemple, c'est n4.

(2) Indique le nom du PVC créé à l'étape précédente. Dans cet exemple, c'est mon-pfc.

c. Confirmez que le déploiement a été créé avec succès en exécutant la commande suivante:

```
$ oc get deployment
```

Exemple de sortie

```
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
postgres  0/1     1            0           42s
```

Il peut prendre quelques minutes pour que les instances d'hyperdisque complètent le provisionnement et affichent un statut READY.

d. Confirmez que le PVC my-pvc a été lié avec succès à un volume persistant (PV) en exécutant la commande suivante:

```
$ oc get pvc my-pvc
```

Exemple de sortie

```
NAME        STATUS   VOLUME                                     CAPACITY   ACCESS MODES
```

```
STORAGECLASS      VOLUMEATTRIBUTESCLASS  AGE
my-pvc      Bound    pvc-1ff52479-4c81-4481-aa1d-b21c8f8860c6   2Ti        RWO
hyperdisk-sc      <unset>                 2m24s
```

e. Confirmez la configuration attendue de votre disque hyperdisque:

```
$ gcloud compute disks list
```

Exemple de sortie

```
NAME                                  LOCATION       LOCATION_SCOPE  SIZE_GB  TYPE
STATUS
instance-20240914-173145-boot         us-central1-a  zone            150      pd-standard
READY
instance-20240914-173145-data-workspace  us-central1-a  zone         100      pd-
balanced        READY
c4a-rhel-vm                           us-central1-a  zone            50       hyperdisk-balanced
READY (1)
```

(1) Disque hyperdisk-équilibré.

f. En utilisant des pools de stockage, vérifiez que le volume est fourni comme spécifié dans votre classe de stockage et PVC en exécutant la commande suivante:

```
$ gcloud compute storage-pools list-disks pool-us-east4-c --zone=us-east4-c
```

Exemple de sortie

```
NAME                                  STATUS  PROVISIONED_IOPS
PROVISIONED_THROUGHPUT  SIZE_GB
pvc-1ff52479-4c81-4481-aa1d-b21c8f8860c6  READY   3000             140
2048
```

5.5.3. À propos de CSI

Les fournisseurs de stockage ont traditionnellement fourni des pilotes de stockage dans le cadre de Kubernetes. Avec l'implémentation de l'interface de stockage de conteneurs (CSI), les fournisseurs tiers peuvent fournir des plugins de stockage à l'aide d'une interface standard sans jamais avoir à modifier le code Kubernetes de base.

Les opérateurs CSI offrent aux utilisateurs d'OpenShift des options de stockage, telles que des instantanés de volume, qui ne sont pas possibles avec les plugins de volume dans l'arbre.

5.5.4. GCP PD CSI paramètres de classe de stockage de pilotes

Le pilote Google Cloud Platform (GCP) Container Storage Interface (PD) de disque persistant (CSI) utilise le sidecar externe CSI comme contrôleur. Il s'agit d'un conteneur d'assistance séparé qui est déployé avec le pilote CSI. Le sidecar gère les volumes persistants (PV) en déclenchant l'opération CreateVolume.

Le pilote GCP PD CSI utilise la clé de paramètre csi.storage.k8s.io/fstype pour prendre en charge le provisionnement dynamique. Le tableau suivant décrit tous les paramètres de la classe de stockage CSI PCP PD qui sont pris en charge par OpenShift Dedicated.

Tableau 5.2. Créer des paramètres de volume

Le paramètre	Les valeurs	Défaut par défaut	Description
le type	B) PD-sd, pd-standard ou pd-équilibré	**la norme PD-standard**	Il vous permet de choisir entre les PV standard ou les PV à l'état solide. Le pilote ne valide pas la valeur, ainsi toutes les valeurs possibles sont acceptées.
le type de réplication	aucun ou régional-pd	**aucune.**	Il vous permet de choisir entre les PV zonaux ou régionaux.
disque-encryption-kms-key	Identifiant de ressource entièrement qualifié pour la clé à utiliser pour chiffrer de nouveaux disques.	Chaîne vide	Il utilise des clés de chiffrement gérées par le client (CMEK) pour chiffrer de nouveaux disques.

5.5.5. Création d'un volume persistant crypté sur mesure

Lorsque vous créez un objet PersistentVolumeClaim, OpenShift Dédie un nouveau volume persistant (PV) et crée un objet PersistentVolume. Il est possible d'ajouter une clé de chiffrement personnalisée dans Google Cloud Platform (GCP) pour protéger un PV dans votre cluster en cryptant le PV nouvellement créé.

En ce qui concerne le chiffrement, le PV nouvellement connecté que vous créez utilise des clés de chiffrement gérées par le client (CMEK) sur un cluster à l'aide d'une clé de gestion de clés Google Cloud (KMS) nouvelle ou existante.

Conditions préalables

- Connectez-vous à un cluster dédié OpenShift en cours d'exécution.
- C'est vous qui avez créé un porte-clés Cloud KMS et une version clé.

En savoir plus sur les ressources CMEK et Cloud KMS, voir Utiliser les clés de chiffrement gérées par le client (CMEK).

Procédure

Afin de créer un PV crypté sur mesure, complétez les étapes suivantes:

1. Créez une classe de stockage avec la clé Cloud KMS. L'exemple suivant permet un provisionnement dynamique des volumes cryptés:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
```

```
  name: csi-gce-pd-cmek
provisioner: pd.csi.storage.gke.io
volumeBindingMode: "WaitForFirstConsumer"
allowVolumeExpansion: true
parameters:
  type: pd-standard
  disk-encryption-kms-key: projects/<key-project-id>/locations/<location>/keyRings/<key-
ring>/cryptoKeys/<key> 1
```

1 Ce champ doit être l'identifiant de ressource pour la clé qui sera utilisée pour chiffrer de nouveaux disques. Les valeurs sont sensibles à la casse. En savoir plus sur la fourniture de valeurs clés d'identification, consultez Retrieving Id d'une ressource et Getting a Cloud KMS ID.

NOTE

Il est impossible d'ajouter le paramètre disque-encryption-kms-key à une classe de stockage existante. Cependant, vous pouvez supprimer la classe de stockage et la recréer avec le même nom et un ensemble différent de paramètres. Dans ce cas, le provisionneur de la classe existante doit être pd.csi.storage.gke.io.

2. Déployez la classe de stockage sur votre cluster dédié OpenShift à l'aide de la commande oc:

```
$ oc describe storageclass csi-gce-pd-cmek
```

Exemple de sortie

```
Name:                  csi-gce-pd-cmek
IsDefaultClass:        No
Annotations:           None
Provisioner:           pd.csi.storage.gke.io
Parameters:            disk-encryption-kms-key=projects/key-project-
id/locations/location/keyRings/ring-name/cryptoKeys/key-name,type=pd-standard
AllowVolumeExpansion:  true
MountOptions:          none
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                none
```

3. Créez un fichier nommé pvc.yaml qui correspond au nom de l'objet de votre classe de stockage que vous avez créé à l'étape précédente:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: podpvc
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: csi-gce-pd-cmek
  resources:
    requests:
      storage: 6Gi
```

NOTE

Lorsque vous avez marqué la nouvelle classe de stockage par défaut, vous pouvez omettre le champ StorageClassName.

4. Appliquez le PVC sur votre cluster:

```
$ oc apply -f pvc.yaml
```

5. Bénéficiez de l'état de votre PVC et vérifiez qu'il est créé et lié à un PV nouvellement fourni:

```
$ oc get pvc
```

Exemple de sortie

```
NAME     STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS   AGE
podpvc   Bound    pvc-e36abf50-84f3-11e8-8538-42010a800002   10Gi       RWO           csi-
gce-pd-cmek  9s
```

NOTE

Lorsque votre classe de stockage a le champ VolumeBindingMode sur WaitForFirstConsumer, vous devez créer un pod pour utiliser le PVC avant de pouvoir le vérifier.

Le PV protégé par CMEK est maintenant prêt à être utilisé avec votre cluster OpenShift Dedicated.

5.5.6. Ressources supplémentaires

- Stockage persistant à l'aide du disque persistant GCE
- Configuration des volumes CSI

5.6. GOOGLE COMPUTE PLATFORM FILESTORE CSI DRIVER OPERATOR

5.6.1. Aperçu général

Le logiciel OpenShift Dedicated est capable de fournir des volumes persistants (PV) à l'aide du pilote CSI (Container Storage Interface) pour Google Compute Platform (GCP) Filestore Storage.

La familiarité avec le stockage persistant et la configuration des volumes CSI est recommandée lorsque vous travaillez avec un opérateur CSI et un conducteur.

Afin de créer des PV fournis par CSI qui sont montés sur les actifs GCP Filestore Storage, vous installez l'opérateur de pilotes GCP Filestore CSI et le pilote GCP Filestore CSI dans l'espace de noms openshift-cluster-csi-drivers.

- Le GCP Filestore CSI Driver Operator ne fournit pas une classe de stockage par défaut, mais vous pouvez en créer une si nécessaire. Le GCP Filestore CSI Driver Operator prend en charge le provisionnement dynamique de volume en permettant la création de volumes de stockage à

la demande, éliminant ainsi la nécessité pour les administrateurs de cluster de pré-fournir le stockage.

- Le pilote GCP Filestore CSI vous permet de créer et de monter des PV GCP Filestore.

La banque de fichiers GCP dédiée à OpenShift prend en charge l'identité de charge de travail. Cela permet aux utilisateurs d'accéder aux ressources Google Cloud en utilisant des identités fédérées au lieu d'une clé de compte de service. GCP Workload Identity doit être activé globalement pendant l'installation, puis configuré pour l'opérateur de pilotes CSI Filestore GCP. Cliquez ici pour plus d'informations sur Instaling the GCP Filestore CSI Driver Operator.

5.6.2. À propos de CSI

Les fournisseurs de stockage ont traditionnellement fourni des pilotes de stockage dans le cadre de Kubernetes. Avec l'implémentation de l'interface de stockage de conteneurs (CSI), les fournisseurs tiers peuvent fournir des plugins de stockage à l'aide d'une interface standard sans jamais avoir à modifier le code Kubernetes de base.

Les opérateurs CSI offrent aux utilisateurs d'OpenShift des options de stockage, telles que des instantanés de volume, qui ne sont pas possibles avec les plugins de volume dans l'arbre.

5.6.3. Installation du gestionnaire de pilotes GCP Filestore CSI

5.6.3.1. Installation du GCP Filestore CSI Driver Operator avec Workload Identity

Lorsque vous envisagez d'utiliser GCP Workload Identity avec Google Compute Platform Filestore, vous devez obtenir certains paramètres que vous utiliserez lors de l'installation de l'opérateur de pilotes GCP Filestore Container Storage Interface (CSI).

Conditions préalables

- Accès au cluster en tant qu'utilisateur avec le rôle cluster-admin.

Procédure

Afin de préparer l'installation de GCP Filestore CSI Driver Operator avec Workload Identity:

1. Obtenir le numéro de projet:
   a. Demandez l'ID du projet en exécutant la commande suivante:

```
$ export PROJECT_ID=$(oc get infrastructure/cluster -o
jsonpath='{.status.platformStatus.gcp.projectID}')
```

   b. Obtenir le numéro de projet, à l'aide de l'ID du projet, en exécutant la commande suivante:

```
$ gcloud projects describe $PROJECT_ID --format="value(projectNumber)"
```

2. Cherchez l'identifiant du pool d'identités et l'ID du fournisseur:
   Lors de l'installation du cluster, les noms de ces ressources sont fournis à l'utilitaire Cloud Credential Operator (ccoctl) avec le paramètre --name. Consultez « Créer des ressources GCP avec l'utilitaire Cloud Credential Operator ».

3. Créer des ressources d'identité de charge de travail pour l'opérateur GCP Filestore:

a. Créez un fichier CredentialsRequest à l'aide de l'exemple suivant:

Exemple d'identification Demandez le fichier YAML

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: openshift-gcp-filestore-csi-driver-operator
  namespace: openshift-cloud-credential-operator
  annotations:
    include.release.openshift.io/self-managed-high-availability: "true"
    include.release.openshift.io/single-node-developer: "true"
spec:
  serviceAccountNames:
  - gcp-filestore-csi-driver-operator
  - gcp-filestore-csi-driver-controller-sa
  secretRef:
    name: gcp-filestore-cloud-credentials
    namespace: openshift-cluster-csi-drivers
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
 kind: GCPProviderSpec
    predefinedRoles:
    - roles/file.editor
    - roles/resourcemanager.tagUser
    skipServiceCheck: true
```

b. Le fichier CredentialsRequest permet de créer un compte de service GCP en exécutant la commande suivante:

```
$ ./ccoctl gcp create-service-accounts --name=<filestore-service-account> \ 1
  --workload-identity-pool=<workload-identity-pool> \ 2
  --workload-identity-provider=<workload-identity-provider> \ 3
  --project=<project-id> \ 4
  --credentials-requests-dir=/tmp/credreq 5
```

1. <Filestore-service-account> est un nom choisi par l'utilisateur.
2. <workload-identity-pool> vient de l'étape 2 ci-dessus.
3. <workload-identity-provider> vient de l'étape 2 ci-dessus.
4. <project-id> vient de l'étape 1.a ci-dessus.
5. Le nom du répertoire où réside le fichier CredentialsRequest.

Exemple de sortie

```
2025/02/10 17:47:39 Credentials loaded from gcloud CLI defaults
2025/02/10 17:47:42 IAM service account filestore-service-account-openshift-gcp-filestore-csi-driver-operator created
2025/02/10 17:47:44 Unable to add predefined roles to IAM service account, retrying...
2025/02/10 17:47:59 Updated policy bindings for IAM service account filestore-service-
```

```
account-openshift-gcp-filestore-csi-driver-operator
2025/02/10 17:47:59 Saved credentials configuration to: /tmp/install-dir/ 1
openshift-cluster-csi-drivers-gcp-filestore-cloud-credentials-credentials.yaml
```

1 Le répertoire actuel.

c. Cherchez l'e-mail du compte de service nouvellement créé en exécutant la commande suivante:

```
$ cat /tmp/install-dir/manifests/openshift-cluster-csi-drivers-gcp-filestore-cloud-credentials-credentials.yaml | yq '.data["service_account.json"]' | base64 -d | jq '.service_account_impersonation_url'
```

Exemple de sortie

```
https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/filestore-se-openshift-g-ch8cm@openshift-gce-devel.iam.gserviceaccount.com:generateAccessToken
```

Dans cet exemple de sortie, l'e-mail du compte de service est filestore-se-openshift-g-ch8cm@openshift-gce-devel.iam.gserviceaccount.com.

Les résultats

Il y a maintenant les paramètres suivants pour installer le GCP Filestore CSI Driver Operator:

- Le numéro du projet - à partir de l'étape 1.b
- ID de piscine - à partir de l'étape 2
- ID du fournisseur - à partir de l'étape 2
- E-mail de compte de service - à partir de l'étape 3.c

5.6.3.2. Installation du gestionnaire de pilotes GCP Filestore CSI

L'opérateur de l'interface de stockage de conteneurs (CSI) Filestore de Google Compute Platform (GCP) n'est pas installé dans OpenShift Dédié par défaut. Dans votre cluster, utilisez la procédure suivante pour installer le GCP Filestore CSI Driver Operator.

Conditions préalables

- Accès à la console Web dédiée OpenShift.
- Lorsque vous utilisez GCP Workload Identity, certains paramètres GCP Workload Identity sont nécessaires. Consultez la section précédente Préparation pour installer l'opérateur de pilotes CSI Filestore GCP avec l'identité de charge de travail.

Procédure

Installer le GCP Filestore CSI Driver Operator à partir de la console web:

1. Connectez-vous à OpenShift Cluster Manager.
2. Choisissez votre cluster.

3. Cliquez sur Ouvrir la console et connectez-vous avec vos informations d'identification.
4. Activer l'API Filestore dans le projet GCE en exécutant la commande suivante:

```
$ gcloud services enable file.googleapis.com  --project <my_gce_project> 1
```

1 <my_gce_project> par votre projet Google Cloud.

Il est également possible de le faire à l'aide de la console Web Google Cloud.

5. Installez l'opérateur GCP Filestore CSI:
   a. Cliquez sur Opérateurs → OperatorHub.
   b. Localisez l'opérateur GCP Filestore CSI en tapant GCP Filestore dans la zone de filtre.
   c. Cliquez sur le bouton GCP Filestore CSI Driver Operator.
   d. Dans la page GCP Filestore CSI Driver Operator, cliquez sur Installer.
   e. Dans la page Installer l'opérateur, assurez-vous que:
      - L'ensemble des espaces de noms du cluster (par défaut) est sélectionné.
      - Installé Namespace est réglé sur openshift-cluster-csi-drivers.
        Lorsque vous utilisez GCP Workload Identity, entrez les valeurs pour les champs suivants obtenus à partir de la procédure dans la section Préparation à l'installation de l'opérateur de pilotes CSI Filestore GCP avec l'identité de charge de travail:
      - **GCP Numéro de projet**
      - **GCP Pool ID**
      - **Carte d'identité du fournisseur GCP**
      - **GCP Service Compte Email**
   f. Cliquez sur **Install**.
      Après la fin de l'installation, l'opérateur GCP Filestore CSI est listé dans la section Opérateurs installés de la console Web.
6. Installez le pilote GCP Filestore CSI:
   a. Cliquez sur administration → CustomResourceDefinitions → ClusterCSIDriver.
   b. Dans l'onglet Instances, cliquez sur Créer un clusterCSIDriver.
      Consultez le fichier YAML suivant:

```
apiVersion: operator.openshift.io/v1
kind: ClusterCSIDriver
metadata:
    name: filestore.csi.storage.gke.io
spec:
  managementState: Managed
```

   c. Cliquez sur **Create**.

d. Attendez que les Conditions suivantes changent à un statut « vrai »:

- GCPFilestoreDriverCredentialsRequestControllerDisponible
- GCPFilestoreDriverNodeServiceControllerAvailable
- GCPFilestoreDriverControllerServiceControllerDisponible

Ressources supplémentaires

- Activer une API dans votre Google Cloud.
- Activer une API à l'aide de la console Web Google Cloud.

5.6.4. Création d'une classe de stockage pour GCP Filestore Storage

Après avoir installé l'opérateur, vous devez créer une classe de stockage pour le provisionnement dynamique des volumes de Google Compute Platform (GCP) Filestore.

Conditions préalables

- Connectez-vous au cluster OpenShift Dédicated en cours d'exécution.

Procédure

Créer une classe de stockage:

1. Créez une classe de stockage à l'aide de l'exemple de fichier YAML suivant:

Exemple de fichier YAML

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: filestore-csi
provisioner: filestore.csi.storage.gke.io
parameters:
  connect-mode: DIRECT_PEERING 1
  network: network-name 2
allowVolumeExpansion: true
volumeBindingMode: WaitForFirstConsumer
```

1 Dans le cas d'un VPC partagé, utilisez le paramètre connect-mode défini sur PRIVATE_SERVICE_ACCESS. Dans le cas d'un VPC non partagé, la valeur est DIRECT_PEERING, qui est le paramètre par défaut.

2 Indiquez le nom du réseau de cloud privé virtuel (VPC) GCP dans lequel les instances Filestore doivent être créées.

2. Indiquez le nom du réseau VPC dans lequel les instances Filestore doivent être créées. Il est recommandé de spécifier le réseau VPC dans lequel les instances Filestore doivent être créées. En l'absence de réseau VPC, le pilote Container Storage Interface (CSI) tente de créer les instances dans le réseau VPC par défaut du projet.

Dans les installations IPI, le nom du réseau VPC est généralement le nom de cluster avec le suffixe "-réseau". Cependant, sur les installations UPI, le nom du réseau VPC peut être n'importe quelle valeur choisie par l'utilisateur.

Dans le cas d'un PCV partagé (mode connexion = PRIVATE_SERVICE_ACCESS), le réseau doit être le nom complet du VPC. Exemple : projets/shared-vpc-name/global/networks/gcp-filestore-network.

Découvrez le nom du réseau VPC en inspectant les objets MachineSets avec la commande suivante:

```
$ oc -n openshift-machine-api get machinesets -o yaml | grep "network:"
            - network: gcp-filestore-network
(...)
```

Dans cet exemple, le nom du réseau VPC dans ce cluster est "gcp-filestore-network".

5.6.5. Destruction des clusters et GCP Filestore

En règle générale, si vous détruisez un cluster, l'installateur OpenShift Dedicated supprime toutes les ressources cloud qui appartiennent à ce cluster. Cependant, en raison de la nature particulière des ressources de Google Compute Platform (GCP) Filestore, le processus de nettoyage automatisé pourrait ne pas les supprimer tous dans de rares cas.

C'est pourquoi Red Hat vous recommande de vérifier que toutes les ressources Filestore détenues par le cluster sont supprimées par le processus de désinstallation.

Procédure

Afin de s'assurer que tous les PVC GCP Filestore ont été supprimés:

1. Accédez à votre compte Google Cloud en utilisant l'interface graphique ou CLI.

2. Cherchez toutes les ressources avec l'étiquette appartenant à kubernetes-io-cluster-${CLUSTER_ID}=.
   Étant donné que l'ID de cluster est unique au cluster supprimé, il ne devrait pas y avoir de ressources restantes avec cet ID de cluster.

3. Dans le cas peu probable, il y a quelques ressources restantes, supprimez-les.

5.6.6. Ressources supplémentaires

- Configuration des volumes CSI

CHAPITRE 6. LES VOLUMES ÉPHÉMÈRES GÉNÉRIQUES

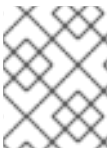
6.1. APERÇU GÉNÉRAL

Les volumes éphémères génériques sont un type de volume éphémère qui peut être fourni par tous les pilotes de stockage qui prennent en charge les volumes persistants et le provisionnement dynamique. Les volumes éphémères génériques sont similaires aux volumes vides en ce sens qu'ils fournissent un répertoire par pod pour les données de grattage, qui est généralement vide après provisionnement.

Les volumes éphémères génériques sont spécifiés en ligne dans le pod spec et suivent le cycle de vie de la gousse. Ils sont créés et supprimés avec le pod.

Les volumes éphémères génériques ont les caractéristiques suivantes:

- Le stockage peut être local ou connecté au réseau.
- Les volumes peuvent avoir une taille fixe que les gousses ne peuvent pas dépasser.
- Les volumes peuvent avoir certaines données initiales, en fonction du pilote et des paramètres.
- Les opérations typiques sur les volumes sont prises en charge, en supposant que le conducteur les supporte, y compris le snapshotting, le clonage, le redimensionnement et le suivi des capacités de stockage.



NOTE

Les volumes éphémères génériques ne prennent pas en charge les instantanés et les redimensionnements hors ligne.

6.2. ALLÉGATIONS DE CYCLE DE VIE ET DE VOLUME PERSISTANT

Les paramètres d'une revendication de volume sont autorisés à l'intérieur d'une source de volume d'un pod. Les étiquettes, les annotations et l'ensemble des champs pour les revendications de volume persistants (PVC) sont pris en charge. Lorsqu'une telle gousse est créée, le contrôleur de volume éphémère crée alors un objet PVC réel (à partir du modèle présenté dans la procédure de création de volumes éphémères génériques) dans le même espace de noms que la gousse, et s'assure que le PVC est supprimé lorsque la gousse est supprimée. Cela déclenche la liaison et l'approvisionnement en volume de l'une des deux façons suivantes:

- Immédiatement, si la classe de stockage utilise une liaison de volume immédiate.
  Avec une liaison immédiate, le planificateur est obligé de sélectionner un nœud qui a accès au volume après qu'il soit disponible.
- Lorsque le pod est provisoirement programmé sur un nœud (mode de liaison WaitForFirstConsumervolume).
  Cette option de liaison de volume est recommandée pour les volumes éphémères génériques car le planificateur peut alors choisir un nœud approprié pour le pod.

En termes de propriété des ressources, un pod qui a un stockage éphémère générique est le propriétaire des PVC qui fournissent ce stockage éphémère. Lorsque le pod est supprimé, le collecteur d'ordures Kubernetes supprime le PVC, ce qui déclenche généralement la suppression du volume parce que la politique de récupération par défaut des classes de stockage est de supprimer les volumes. Il est possible de créer un stockage local quasi éphémère en utilisant une classe de stockage avec une politique de récupération de conservation: le stockage survit à la pod, et dans ce cas, vous devez vous assurer que le nettoyage du volume se déroule séparément. Bien que ces PVC existent, ils peuvent être

utilisés comme n'importe quel autre PVC. En particulier, ils peuvent être référencés comme source de données dans le clonage de volume ou le snapshotting. L'objet PVC détient également l'état actuel du volume.

Ressources supplémentaires

- Créer des volumes éphémères génériques

6.3. LA SÉCURITÉ

Il est possible d'activer la fonction générique de volume éphémère pour permettre aux utilisateurs qui peuvent créer des pods de créer indirectement des revendications de volume persistantes (PVC). Cette fonctionnalité fonctionne même si ces utilisateurs n'ont pas la permission de créer des PVC directement. Les administrateurs de clusters doivent en être conscients. Dans le cas où cela ne correspond pas à votre modèle de sécurité, utilisez un webhook d'admission qui rejette des objets tels que des pods ayant un volume éphémère générique.

Le quota d'espace de noms normal pour les PVC s'applique toujours, donc même si les utilisateurs sont autorisés à utiliser ce nouveau mécanisme, ils ne peuvent pas l'utiliser pour contourner d'autres politiques.

6.4. DÉNOMINATION PERSISTANTE DES REVENDICATIONS DE VOLUME

Les claims de volume persistants (PVC) créés automatiquement sont nommés par une combinaison du nom du pod et du nom de volume, avec un trait d'union (-) au milieu. Cette convention de dénomination introduit également un conflit potentiel entre les différentes gousses, et entre les gousses et les PVC créés manuellement.

À titre d'exemple, pod-a avec égratignure de volume et gousse avec volume a-cratch se retrouvent tous deux avec le même nom de PVC, pod-a-scratch.

De tels conflits sont détectés et un PVC n'est utilisé que pour un volume éphémère s'il a été créé pour la gousse. Ce contrôle est basé sur la relation de propriété. Le PVC existant n'est pas écrasé ou modifié, mais cela ne résout pas le conflit. En l'absence du PVC droit, une gousse ne peut pas démarrer.

IMPORTANT

Faites attention lorsque vous nommez des pods et des volumes à l'intérieur du même espace de noms afin que les conflits de dénomination ne se produisent pas.

6.5. CRÉER DES VOLUMES ÉPHÉMÈRES GÉNÉRIQUES

Procédure

1. Créez la définition de l'objet pod et enregistrez-le dans un fichier.
2. Inclure les informations génériques sur le volume éphémère dans le fichier.

   exemple-pod-with-generic-vols.yaml

   ```
   kind: Pod
   apiVersion: v1
   ```

```
metadata:
  name: my-app
spec:
  containers:
    - name: my-frontend
      image: busybox:1.28
      volumeMounts:
      - mountPath: "/mnt/storage"
        name: data
      command: [ "sleep", "1000000" ]
  volumes:
    - name: data (1)
      ephemeral:
        volumeClaimTemplate:
          metadata:
            labels:
              type: my-app-ephvol
          spec:
            accessModes: [ "ReadWriteOnce" ]
            storageClassName: "gp2-csi"
            resources:
              requests:
                storage: 1Gi
```

(1) Revendication de volume éphémère générique.

CHAPITRE 7. APPROVISIONNEMENT DYNAMIQUE

7.1. À PROPOS DU PROVISIONNEMENT DYNAMIQUE

L'objet de ressource StorageClass décrit et classe le stockage qui peut être demandé, ainsi que fournit un moyen de passer les paramètres pour le stockage dynamiquement provisionné à la demande. Les objets Storageclass peuvent également servir de mécanisme de gestion pour contrôler différents niveaux de stockage et l'accès au stockage. Les administrateurs de cluster (cluster-admin) ou les administrateurs de stockage (stockage-admin) définissent et créent les objets StorageClass que les utilisateurs peuvent demander sans avoir besoin de connaissances détaillées sur les sources de volume de stockage sous-jacentes.

Le cadre de volume persistant OpenShift Dedicated permet cette fonctionnalité et permet aux administrateurs de fournir un cluster avec un stockage persistant. Le cadre donne également aux utilisateurs un moyen de demander ces ressources sans avoir aucune connaissance de l'infrastructure sous-jacente.

De nombreux types de stockage sont disponibles pour une utilisation en tant que volumes persistants dans OpenShift Dedicated. Bien qu'ils puissent tous être fournis statiquement par un administrateur, certains types de stockage sont créés dynamiquement à l'aide des API du fournisseur intégré et du plugin.

7.2. PLUGINS DE PROVISIONNEMENT DYNAMIQUE DISPONIBLES

Le logiciel OpenShift Dedicated fournit les plugins de provisionner suivants, qui ont des implémentations génériques pour le provisionnement dynamique qui utilisent l'API du fournisseur configuré du cluster pour créer de nouvelles ressources de stockage:

Le type de stockage	Le nom du plugin provisionner	Les notes
Amazon Elastic Block Store (Amazon EBS)	**Kubernetes.io/aws-ebs**	Lorsque vous utilisez plusieurs clusters dans différentes zones, balisez chaque nœud avec Key=kubernetes.io/cluster/<cluster_name>, Value=<cluster_id> où <cluster_name> et <cluster_id> sont uniques par cluster.
Disque persistant GCE (gcePD)	**Kubernetes.io/gce-pd**	Dans les configurations multi-zones, il est conseillé d'exécuter un cluster OpenShift dédié par projet GCE pour éviter que les PV ne soient créés dans des zones où il n'existe aucun nœud dans le cluster actuel.

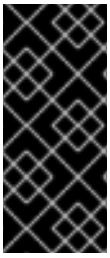
Le type de stockage	Le nom du plugin provisionner	Les notes
Bloc de serveur virtuel IBM Power®	**accueil > Powervs.csi.ibm.com**	Après l'installation, IBM Power® Virtual Server Block CSI Driver Operator et IBM Power® Virtual Server Block CSI Driver créent automatiquement les classes de stockage requises pour le provisionnement dynamique.

IMPORTANT

Le plugin de provisionneur choisi nécessite également une configuration pour le cloud, l'hôte ou le fournisseur tiers concerné conformément à la documentation pertinente.

7.3. DÉFINIR UNE CLASSE DE STOCKAGE

Les objets Storageclass sont actuellement un objet à portée mondiale et doivent être créés par des utilisateurs de cluster-admin ou de stockage-admin.



IMPORTANT

L'opérateur de stockage du cluster peut installer une classe de stockage par défaut en fonction de la plate-forme utilisée. Cette classe de stockage est détenue et contrôlée par l'opérateur. Il ne peut pas être supprimé ou modifié au-delà de la définition des annotations et des étiquettes. Lorsqu'un comportement différent est souhaité, vous devez définir une classe de stockage personnalisée.

Les sections suivantes décrivent la définition de base d'un objet StorageClass et des exemples spécifiques pour chacun des types de plugin pris en charge.

7.3.1. Définition d'objet Basic StorageClass

La ressource suivante affiche les paramètres et les valeurs par défaut que vous utilisez pour configurer une classe de stockage. Cet exemple utilise la définition d'objet AWS ElasticBlockStore (EBS).

Définition de classe de stockage d'échantillons

```
kind: StorageClass (1)
apiVersion: storage.k8s.io/v1 (2)
metadata:
  name: <storage-class-name> (3)
  annotations: (4)
    storageclass.kubernetes.io/is-default-class: 'true'
    ...
provisioner: kubernetes.io/aws-ebs (5)
parameters: (6)
  type: gp3
...
```

1. (obligatoire) Le type d'objet API.
2. (obligatoire) L'apiVersion actuelle.
3. (obligatoire) Le nom de la classe de stockage.
4. (facultatif) Annotations pour la classe de stockage.
5. (obligatoire) Le type de provisionneur associé à cette classe de stockage.
6. (facultatif) Les paramètres requis pour le provisionneur spécifique, cela passera du plug-in au plug-in.

7.3.2. Annotations de classe de stockage

Afin de définir une classe de stockage en tant que par défaut à l'échelle du cluster, ajoutez l'annotation suivante à vos métadonnées de classe de stockage:

```
storageclass.kubernetes.io/is-default-class: "true"
```

À titre d'exemple:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
...
```

Cela permet à toute revendication de volume persistant (PVC) qui ne spécifie pas une classe de stockage spécifique d'être automatiquement fournie par l'intermédiaire de la classe de stockage par défaut. Cependant, votre cluster peut avoir plus d'une classe de stockage, mais un seul d'entre eux peut être la classe de stockage par défaut.

NOTE

La bêta annotation storageclass.beta.kubernetes.io/is-default-class fonctionne toujours; cependant, il sera supprimé dans une version ultérieure.

Afin de définir une description de classe de stockage, ajoutez l'annotation suivante à vos métadonnées de classe de stockage:

```
kubernetes.io/description: My Storage Class Description
```

À titre d'exemple:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    kubernetes.io/description: My Storage Class Description
...
```

7.3.3. Définition d'objet AWS Elastic Block Store (EBS)

AWS-ebs-storageclass.yaml

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: <storage-class-name> 1
provisioner: kubernetes.io/aws-ebs
parameters:
  type: io1 2
  iopsPerGB: "10" 3
  encrypted: "true" 4
  kmsKeyId: keyvalue 5
  fsType: ext4 6
```

1. (obligatoire) Nom de la classe de stockage. La revendication de volume persistant utilise cette classe de stockage pour approvisionner les volumes persistants associés.
2. (obligatoire) Sélectionnez parmi io1, gp3, sc1, st1. La valeur par défaut est gp3. Consultez la documentation AWS pour les valeurs Amazon Resource Name (ARN) valides.
3. Facultatif: Seulement pour les volumes io1. E/S opérations par seconde par GiB. Le plugin de volume AWS multiplie cela avec la taille du volume demandé pour calculer l'IOPS du volume. Le plafond de valeur est de 20 000 IOPS, ce qui est le maximum pris en charge par AWS. Consultez la documentation AWS pour plus de détails.
4. Facultatif: Dénote s'il faut chiffrer le volume EBS. Les valeurs valides sont vraies ou fausses.
5. Facultatif: L'ARN complet de la clé à utiliser lors du chiffrement du volume. Dans le cas où aucun n'est fourni, mais encypted est réglé sur true, AWS génère une clé. Consultez la documentation AWS pour obtenir une valeur ARN valide.
6. Facultatif: Système de fichiers qui est créé sur des volumes provisionnés dynamiquement. Cette valeur est copiée dans le champ fsType des volumes persistants provisionnés dynamiquement et le système de fichiers est créé lorsque le volume est monté pour la première fois. La valeur par défaut est ext4.

7.3.4. GCE PersistentDisk (gcePD) définition d'objet

GCE-pd-storageclass.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <storage-class-name> 1
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd 2
  replication-type: none
volumeBindingMode: WaitForFirstConsumer
allowVolumeExpansion: true
reclaimPolicy: Delete
```

1. Le nom de la classe de stockage. La revendication de volume persistant utilise cette classe de stockage pour approvisionner les volumes persistants associés.
2. Choisissez pd-ssd, pd-standard ou hyperdisk-balanced. La valeur par défaut est pd-ssd.

7.4. CHANGER LA CLASSE DE STOCKAGE PAR DÉFAUT

La procédure suivante permet de modifier la classe de stockage par défaut.

À titre d'exemple, si vous avez deux classes de stockage définies, gp3 et standard, et que vous souhaitez changer la classe de stockage par défaut de gp3 à standard.

Conditions préalables

- Accès au cluster avec des privilèges cluster-admin.

Procédure

Changer la classe de stockage par défaut:

1. Liste des classes de stockage:

```
$ oc get storageclass
```

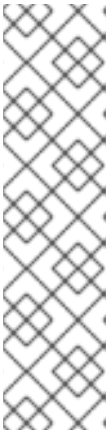
Exemple de sortie

```
NAME            TYPE
gp3 (default)    kubernetes.io/aws-ebs 1
standard        kubernetes.io/aws-ebs
```

1. (par défaut) indique la classe de stockage par défaut.

2. Faites de la classe de stockage souhaitée la valeur par défaut.
Dans la classe de stockage souhaitée, définissez l'annotation de classe storageclass.kubernetes.io/is-default-class en exécutant la commande suivante:

```
$ oc patch storageclass standard -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class": "true"}}}'
```



NOTE

Il est possible d'avoir plusieurs classes de stockage par défaut pendant une courte période. Cependant, vous devez vous assurer qu'une seule classe de stockage par défaut existe éventuellement.

Avec plusieurs classes de stockage par défaut présentes, toute revendication de volume persistant (PVC) demandant la classe de stockage par défaut (pvc.spec.storageClassName=nil) obtient la classe de stockage par défaut la plus récente, quel que soit l'état par défaut de cette classe de stockage, et l'administrateur reçoit une alerte dans le tableau de bord des alertes qu'il existe plusieurs classes de stockage par défaut, MultipleDefaultStorageClasses.

3. Supprimez le paramètre de classe de stockage par défaut de l'ancienne classe de stockage par défaut.
   Dans le cas de l'ancienne classe de stockage par défaut, modifiez la valeur de l'annotation de la classe storageclass.kubernetes.io/is-default-class en exécutant la commande suivante:

```
$ oc patch storageclass gp3 -p '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class": "false"}}}'
```

4. Vérifier les changements:

```
$ oc get storageclass
```

Exemple de sortie

```
NAME                 TYPE
gp3                  kubernetes.io/aws-ebs
standard (default)   kubernetes.io/aws-ebs
```