Plate-forme de conteneurs OpenShift 4.12

Registre

Configuration des registres pour OpenShift Container Platform

Last Updated: 2023-04-26

Plate-forme de conteneurs OpenShift 4.12 Registre

Configuration des registres pour OpenShift Container Platform

Notice légale

Résumé

Ce document fournit des instructions pour configurer et gérer le registre interne d'OpenShift Container Platform. Il donne également un aperçu général des registres associés à OpenShift Container Platform.

Table des matières

CHAPITRE 1. APERÇU DU REGISTRE OPENSHIFT CONTAINER PLATFORM

OpenShift Container Platform peut construire des images à partir de votre code source, les déployer et gérer leur cycle de vie. Elle fournit un registre d'images de conteneurs interne et intégré qui peut être déployé dans votre environnement OpenShift Container Platform pour gérer localement les images. Cette vue d'ensemble contient des informations de référence et des liens pour les registres couramment utilisés avec OpenShift Container Platform, en mettant l'accent sur le registre d'images interne.

1.1. GLOSSAIRE DES TERMES COURANTS POUR LE REGISTRE DE OPENSHIFT CONTAINER PLATFORM

Ce glossaire définit les termes communs utilisés dans le contenu du registre.

conteneur

Images légères et exécutables composées d'un logiciel et de toutes ses dépendances. Comme les conteneurs virtualisent le système d'exploitation, vous pouvez les exécuter dans un centre de données, un nuage public ou privé, ou votre hôte local.

Opérateur de registre d'images

L'opérateur de registre d'images fonctionne dans l'espace de noms **openshift-image-registry** et gère l'instance de registre à cet endroit.

dépôt d'images

Un référentiel d'images est une collection d'images de conteneurs apparentées et d'étiquettes identifiant les images.

registre miroir

Le registre miroir est un registre qui contient le miroir des images d'OpenShift Container Platform.

espace de noms

Un espace de noms isole des groupes de ressources au sein d'un même cluster.

Registre OpenShift Container Platform

OpenShift Container Platform registry est le registre fourni par OpenShift Container Platform pour gérer les images.

nacelle

Le pod est la plus petite unité logique de Kubernetes. Un pod est composé d'un ou plusieurs conteneurs à exécuter dans un worker node.

registre privé

Un registre est un serveur qui met en œuvre l'API de registre d'images de conteneurs. Un registre privé est un registre qui nécessite une authentification pour permettre aux utilisateurs d'accéder à son contenu.

registre public

Un registre est un serveur qui met en œuvre l'API de registre d'images de conteneurs. Un registre public est un registre dont le contenu est accessible au public.

Quay.io

Une instance publique de Red Hat Quay Container Registry fournie et maintenue par Red Hat, qui sert la plupart des images de conteneurs et des opérateurs aux clusters OpenShift Container Platform.

authentification du registre

Pour pousser et tirer des images vers et depuis des dépôts d'images privés, le registre doit authentifier ses utilisateurs à l'aide d'informations d'identification.

itinéraire

Expose un service pour permettre l'accès réseau aux pods à partir d'utilisateurs et d'applications en dehors de l'instance OpenShift Container Platform.

réduire

Pour diminuer le nombre de répliques.

augmenter

Pour augmenter le nombre de répliques.

service

Un service expose une application en cours d'exécution sur un ensemble de pods.

1.2. REGISTRE INTÉGRÉ D'OPENSHIFT CONTAINER PLATFORM

OpenShift Container Platform fournit un registre d'images de conteneurs intégré qui s'exécute comme une charge de travail standard sur le cluster. Le registre est configuré et géré par un opérateur d'infrastructure. Il s'agit d'une solution prête à l'emploi qui permet aux utilisateurs de gérer les images qui exécutent leurs charges de travail, et qui s'exécute au-dessus de l'infrastructure de cluster existante. Ce registre peut être augmenté ou réduit comme n'importe quelle autre charge de travail en cluster et ne nécessite pas de provisionnement spécifique de l'infrastructure. En outre, il est intégré au système d'authentification et d'autorisation des utilisateurs de la grappe, ce qui signifie que l'accès à la création et à l'extraction d'images est contrôlé par la définition des autorisations des utilisateurs sur les ressources d'image.

Le registre est généralement utilisé comme cible de publication pour les images construites sur le cluster, ainsi que comme source d'images pour les charges de travail fonctionnant sur le cluster. Lorsqu'une nouvelle image est envoyée au registre, le cluster en est informé et d'autres composants peuvent réagir et consommer l'image mise à jour.

Les données d'image sont stockées à deux endroits. Les données d'image proprement dites sont stockées dans un emplacement de stockage configurable, tel que le stockage en nuage ou un volume de système de fichiers. Les métadonnées d'image, qui sont exposées par les API standard du cluster et utilisées pour effectuer le contrôle d'accès, sont stockées en tant que ressources API standard, en particulier les images et les flux d'images.

Ressources complémentaires

- Opérateur de registre d'images dans OpenShift Container Platform

1.3. REGISTRES DE TIERS

OpenShift Container Platform peut créer des conteneurs en utilisant des images provenant de registres tiers, mais il est peu probable que ces registres offrent le même support de notification d'image que le registre intégré d'OpenShift Container Platform. Dans ce cas, OpenShift Container Platform récupère les tags du registre distant lors de la création du flux d'images. Pour actualiser les balises récupérées, exécutez **oc import-image <stream>**. Lorsque de nouvelles images sont détectées, les réactions de construction et de déploiement décrites précédemment se produisent.

1.3.1. Authentification

OpenShift Container Platform peut communiquer avec des registres pour accéder à des dépôts d'images privés en utilisant les informations d'identification fournies par l'utilisateur. Cela permet à OpenShift Container Platform de pousser et de tirer des images vers et depuis des dépôts privés.

1.3.1.1. Authentification du registre avec Podman

Certains registres d'images de conteneurs requièrent une autorisation d'accès. Podman est un outil open source permettant de gérer les conteneurs et les images de conteneurs et d'interagir avec les registres d'images. Vous pouvez utiliser Podman pour authentifier vos informations d'identification, extraire l'image du registre et stocker les images locales dans un système de fichiers local. Voici un exemple générique d'authentification du registre avec Podman.

Procédure

1. Utilisez le catalogue de l'écosystème Red Hat pour rechercher des images de conteneurs spécifiques à partir du référentiel Red Hat et sélectionnez l'image requise.
2. Cliquez sur **Get this image** pour trouver la commande correspondant à votre image de conteneur.
3. Connectez-vous en exécutant la commande suivante et en saisissant votre nom d'utilisateur et votre mot de passe pour vous authentifier :

```
$ podman login registry.redhat.io
 Username:<your_registry_account_username>
 Password:<your_registry_account_password>
```

4. Téléchargez l'image et enregistrez-la localement en exécutant la commande suivante :

```
$ podman pull registry.redhat.io/<repository_name>
```

1.4. REGISTRES RED HAT QUAY

Si vous avez besoin d'un registre d'images de conteneurs de qualité professionnelle, Red Hat Quay est disponible à la fois en tant que service hébergé et en tant que logiciel que vous pouvez installer dans votre propre centre de données ou environnement cloud. Les fonctionnalités avancées du registre de Red Hat Quay comprennent la géo-réplication, le balayage d'images et la possibilité de revenir en arrière sur les images.

Visitez le site Quay.io pour créer votre propre compte de registre Quay hébergé. Ensuite, suivez le tutoriel Quay pour vous connecter au registre Quay et commencer à gérer vos images.

Vous pouvez accéder à votre registre Red Hat Quay depuis OpenShift Container Platform comme n'importe quel registre d'image de conteneur distant.

Ressources complémentaires

- Documentation du produit Red Hat Quay

1.5. AUTHENTIFICATION ACTIVÉE REGISTRE RED HAT

Toutes les images de conteneurs disponibles dans la section Images de conteneurs du catalogue de l'écosystème Red Hat sont hébergées sur un registre d'images, **registry.redhat.io**.

Le registre, **registry.redhat.io**, nécessite une authentification pour l'accès aux images et au contenu hébergé sur OpenShift Container Platform. Après le passage au nouveau registre, le registre existant sera disponible pendant un certain temps.

NOTE

OpenShift Container Platform extrait des images de **registry.redhat.io**, vous devez donc configurer votre cluster pour l'utiliser.

Le nouveau registre utilise les mécanismes OAuth standard pour l'authentification, avec les méthodes suivantes :

- **Authentication token.** Les jetons, générés par les administrateurs, sont des comptes de service qui permettent aux systèmes de s'authentifier auprès du registre des images de conteneurs. Les comptes de service ne sont pas affectés par les modifications apportées aux comptes d'utilisateur, de sorte que la méthode d'authentification par jeton est fiable et résiliente. Il s'agit de la seule option d'authentification prise en charge pour les clusters de production.

- **Web username and password.** Il s'agit de l'ensemble standard d'informations d'identification que vous utilisez pour vous connecter à des ressources telles que **access.redhat.com**. Bien qu'il soit possible d'utiliser cette méthode d'authentification avec OpenShift Container Platform, elle n'est pas prise en charge pour les déploiements de production. Limitez cette méthode d'authentification aux projets autonomes en dehors d'OpenShift Container Platform.

Vous pouvez utiliser **podman login** avec vos informations d'identification, soit votre nom d'utilisateur et votre mot de passe, soit un jeton d'authentification, pour accéder au contenu du nouveau registre.

Tous les flux d'images pointent vers le nouveau registre, qui utilise le secret d'extraction de l'installation pour s'authentifier.

Vous devez placer vos justificatifs dans l'un des endroits suivants :

- **openshift namespace**. Vos informations d'identification doivent exister dans l'espace de noms **openshift** pour que les flux d'images de l'espace de noms **openshift** puissent être importés.

- **Your host**. Vos informations d'identification doivent exister sur votre hôte, car Kubernetes utilise les informations d'identification de votre hôte pour extraire des images.

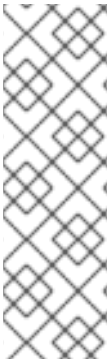
Ressources complémentaires

- Comptes de service du registre

CHAPITRE 2. OPÉRATEUR DE REGISTRE D'IMAGES DANS OPENSHIFT CONTAINER PLATFORM

2.1. REGISTRE D'IMAGES SUR LES PLATEFORMES EN NUAGE ET OPENSTACK

L'opérateur de registre d'images installe une instance unique du registre OpenShift Container Platform et gère toute la configuration du registre, y compris la mise en place du stockage du registre.



NOTE

Le stockage n'est configuré automatiquement que lorsque vous installez un cluster d'infrastructure fourni par l'installateur sur AWS, GCP, Azure ou OpenStack.

Lorsque vous installez ou mettez à niveau un cluster d'infrastructure fourni par l'installateur sur AWS ou Azure, l'opérateur de registre d'images définit le paramètre **spec.storage.managementState** sur **Managed**. Si le paramètre **spec.storage.managementState** est défini sur **Unmanaged**, l'opérateur de registre d'images n'effectue aucune action liée au stockage.

Après le déploiement du plan de contrôle, l'opérateur crée une instance de ressource **configs.imageregistry.operator.openshift.io** par défaut sur la base de la configuration détectée dans le cluster.

Si les informations disponibles sont insuffisantes pour définir une ressource **configs.imageregistry.operator.openshift.io** complète, la ressource incomplète est définie et l'opérateur met à jour l'état de la ressource en indiquant les informations manquantes.

L'opérateur de registre d'images s'exécute dans l'espace de noms **openshift-image-registry** et gère l'instance de registre dans cet emplacement également. Toutes les ressources de configuration et de charge de travail pour le registre résident dans cet espace de noms.

IMPORTANT

managementState Le comportement de l'opérateur de registre d'images pour la gestion de l'élagueur est orthogonal à celui spécifié sur l'objet **ClusterOperator** pour l'opérateur de registre d'images. Si l'opérateur de registre d'images n'est pas dans l'état **Managed**, l'élagueur d'images peut toujours être configuré et géré par la ressource personnalisée **Pruning**.

Cependant, le site **managementState** de l'opérateur de registre d'images modifie le comportement de la tâche d'élagage d'images déployée :

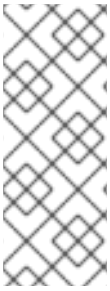
- **Managed**le code **--prune-registry** pour l'élagueur d'images est défini sur **true**.
- **Removed**le drapeau **--prune-registry** de l'élagueur d'images est fixé à **false**, ce qui signifie qu'il n'élague que les métadonnées d'images dans etcd.
- **Unmanaged**le code **--prune-registry** pour l'élagueur d'images est défini sur **false**.

2.2. REGISTRE D'IMAGES SUR BARE METAL ET VSPHERE

2.2.1. Registre d'images supprimé lors de l'installation

Sur les plateformes qui ne fournissent pas de stockage d'objets partageables, l'opérateur de registre d'images OpenShift s'amorce lui-même en tant que **Removed**. Cela permet à **openshift-installer** de réaliser des installations sur ces types de plateformes.

Après l'installation, vous devez modifier la configuration de l'opérateur du registre des images pour faire passer le site **managementState** de **Removed** à **Managed**.



NOTE

La console Prometheus fournit une alerte **ImageRegistryRemoved**, par exemple :

"Image Registry has been removed. **ImageStreamTags** il se peut que les fichiers **BuildConfigs** et **DeploymentConfigs** qui font référence à **ImageStreamTags** ne fonctionnent pas comme prévu. Veuillez configurer le stockage et mettre à jour la configuration à l'état **Managed** en éditant configs.imageregistry.operator.openshift.io."

2.3. DISTRIBUTION DE L'OPÉRATEUR DU REGISTRE D'IMAGES DANS LES ZONES DE DISPONIBILITÉ

La configuration par défaut de l'opérateur de registre d'images répartit les pods de registre d'images dans les zones topologiques afin d'éviter les délais de récupération en cas de défaillance complète d'une zone où tous les pods sont touchés.

L'opérateur de registre d'images prend par défaut les valeurs suivantes lorsqu'il est déployé avec une contrainte topologique liée à une zone :

Opérateur de registre d'images déployé avec une contrainte topologique liée à la zone

```
topologySpreadConstraints:
- labelSelector:
    matchLabels:
      docker-registry: default
  maxSkew: 1
  topologyKey: kubernetes.io/hostname
  whenUnsatisfiable: DoNotSchedule
- labelSelector:
    matchLabels:
      docker-registry: default
  maxSkew: 1
  topologyKey: node-role.kubernetes.io/worker
  whenUnsatisfiable: DoNotSchedule
- labelSelector:
    matchLabels:
      docker-registry: default
  maxSkew: 1
  topologyKey: topology.kubernetes.io/zone
  whenUnsatisfiable: DoNotSchedule
```

L'opérateur de registre d'images est défini par défaut comme suit lorsqu'il est déployé sans contrainte topologique liée à une zone, ce qui s'applique aux instances bare metal et vSphere :

Image Registry Operator déployé sans contrainte topologique liée à une zone

-

```
topologySpreadConstraints:
 - labelSelector:
     matchLabels:
       docker-registry: default
   maxSkew: 1
   topologyKey: kubernetes.io/hostname
   whenUnsatisfiable: DoNotSchedule
 - labelSelector:
     matchLabels:
       docker-registry: default
   maxSkew: 1
   topologyKey: node-role.kubernetes.io/worker
   whenUnsatisfiable: DoNotSchedule
```

Un administrateur de cluster peut remplacer la valeur par défaut de **topologySpreadConstraints** en configurant le fichier spec de **configs.imageregistry.operator.openshift.io/cluster**. Dans ce cas, seules les contraintes que vous fournissez s'appliquent.

2.4. RESSOURCES COMPLÉMENTAIRES

- Configuration des contraintes d'étalement de la topologie des pods

2.5. PARAMÈTRES DE CONFIGURATION DE L'OPÉRATEUR DE REGISTRE D'IMAGES

La ressource **configs.imageregistry.operator.openshift.io** offre les paramètres de configuration suivants.

Paramètres	Description
managementState	**Managed**: L'opérateur met à jour le registre au fur et à mesure que les ressources de configuration sont mises à jour. **Unmanaged**: L'opérateur ignore les modifications apportées aux ressources de configuration. **Removed**: L'opérateur supprime l'instance de registre et démantèle tout stockage qu'il a provisionné.
logLevel	Définit **logLevel** de l'instance de registre. La valeur par défaut est**Normal**. Les valeurs prises en charge pour **logLevel** sont les suivantes : • **Normal** • **Debug** • **Trace** • **TraceAll**
httpSecret	Valeur nécessaire au registre pour sécuriser les téléchargements, générée par défaut.

Paramètres	Description
proxy	Définit le mandataire à utiliser lors de l'appel de l'API principale et des registres en amont.
storage	**Storagetype**: Détails pour configurer le stockage du registre, par exemple les coordonnées d'un seau S3. Normalement configuré par défaut.
readOnly	Indique si l'instance de registre doit rejeter les tentatives d'ajout de nouvelles images ou de suppression d'images existantes.
requests	API Request Limit (Limite de requêtes API). Contrôle le nombre de demandes parallèles qu'une instance de registre donnée peut traiter avant de mettre en file d'attente les demandes supplémentaires.
defaultRoute	Détermine si une route externe est définie ou non à l'aide du nom d'hôte par défaut. Si cette option est activée, l'itinéraire utilise le cryptage re-encrypt. La valeur par défaut est **false**.
routes	Tableau d'itinéraires supplémentaires à créer. Vous fournissez le nom d'hôte et le certificat de l'itinéraire.
rolloutStrategy	Définit la stratégie de déploiement du registre d'images. La valeur par défaut est **RollingUpdate**.
replicas	Nombre de répliques pour le registre.
disableRedirect	Indique si toutes les données doivent passer par le registre, plutôt que d'être redirigées vers le back-end. La valeur par défaut est **false**.

Paramètres	Description
spec.storage.managementState	L'opérateur de registre d'images définit le paramètre **spec.storage.managementState** sur **Managed** lors de nouvelles installations ou de mises à niveau de clusters utilisant une infrastructure fournie par l'installateur sur AWS ou Azure. • **Managed**: Détermine si l'opérateur de registre d'images gère le stockage sous-jacent. Si l'opérateur de registre d'images **managementState** est défini sur **Removed**, le stockage est supprimé. ◦ Si **managementState** est défini sur **Managed**, l'opérateur de registre d'images tente d'appliquer une configuration par défaut à l'unité de stockage sous-jacente. Par exemple, s'il est défini sur **Managed**, l'opérateur tente d'activer le cryptage sur le seau S3 avant de le mettre à la disposition du registre. Si vous ne souhaitez pas que les paramètres par défaut soient appliqués à l'unité de stockage que vous fournissez, assurez-vous que **managementState** est défini sur **Unmanaged**. • **Unmanaged**: Détermine si l'opérateur de registre d'images ignore les paramètres de stockage. Si la valeur **managementState** de l'opérateur de registre d'images est **Removed**, le stockage n'est pas supprimé. Si vous avez fourni une configuration d'unité de stockage sous-jacente, telle qu'un nom de godet ou de conteneur, et que **spec.storage.managementState** n'est pas encore défini sur une valeur quelconque, l'opérateur de registre d'images le configure sur **Unmanaged**.

2.6. ACTIVER LA ROUTE PAR DÉFAUT DU REGISTRE DES IMAGES AVEC LA DÉFINITION DE RESSOURCE PERSONNALISÉE

Dans OpenShift Container Platform, l'opérateur **Registry** contrôle la fonction de registre. L'opérateur est défini par **configs.imageregistry.operator.openshift.io** Custom Resource Definition (CRD).

Si vous avez besoin d'activer automatiquement la route par défaut du registre d'images, patchez le CRD de l'opérateur du registre d'images.

Procédure

- Patch de l'opérateur du registre des images CRD :

```
$ oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec":
{"defaultRoute":true}}'
```

2.7. CONFIGURATION DE MAGASINS DE CONFIANCE SUPPLÉMENTAIRES POUR L'ACCÈS AU REGISTRE D'IMAGES

La ressource personnalisée **image.config.openshift.io/cluster** peut contenir une référence à une carte de configuration qui contient des autorités de certification supplémentaires à approuver lors de l'accès au registre d'images.

Conditions préalables

- Les autorités de certification (CA) doivent être codées en PEM.

Procédure

Vous pouvez créer une carte de configuration dans l'espace de noms **openshift-config** et utiliser son nom dans **AdditionalTrustedCA** dans la ressource personnalisée **image.config.openshift.io** pour fournir des autorités de certification supplémentaires qui doivent être approuvées lorsqu'elles contactent des registres externes.

La clé de la carte de configuration est le nom d'hôte d'un registre avec le port pour lequel cette autorité de certification doit être approuvée, et le certificat codé en base64 est la valeur, pour chaque autorité de certification de registre supplémentaire à approuver.

Registre d'images Exemple de carte de configuration de l'autorité de certification

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-registry-ca
data:
  registry.example.com: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
  registry-with-port.example.com..5000: | 1
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

1 Si le registre comporte le port, tel que **registry-with-port.example.com:5000**, **:** doit être remplacé par **..**.

Vous pouvez configurer des autorités de certification supplémentaires en suivant la procédure suivante.

1. Pour configurer une autorité de certification supplémentaire :

```
$ oc create configmap registry-config --from-file=<external_registry_address>=ca.crt -n
openshift-config
```

```
$ oc edit image.config.openshift.io cluster
```

```
spec:
  additionalTrustedCA:
    name: registry-config
```

2.8. CONFIGURATION DES IDENTIFIANTS DE STOCKAGE POUR L'OPÉRATEUR DE REGISTRE D'IMAGES

Outre les ressources **configs.imageregistry.operator.openshift.io** et ConfigMap, la configuration des justificatifs de stockage est fournie à l'opérateur par une ressource secrète distincte située dans l'espace de noms **openshift-image-registry**.

Le secret **image-registry-private-configuration-user** fournit les informations d'identification nécessaires à l'accès au stockage et à sa gestion. Il remplace les informations d'identification par défaut utilisées par l'opérateur, si des informations d'identification par défaut ont été trouvées.

Procédure

- Créez un secret OpenShift Container Platform qui contient les clés requises.

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=KEY1=value1 --from-literal=KEY2=value2 --namespace openshift-image-registry
```

2.9. RESSOURCES COMPLÉMENTAIRES

- Configuration du registre pour l'infrastructure AWS fournie par l'utilisateur
- Configuration du registre pour l'infrastructure fournie par l'utilisateur GCP
- Configuration du registre pour l'infrastructure Azure fournie par l'utilisateur
- Configuration du registre pour le métal nu
- Configuration du registre pour vSphere

CHAPITRE 3. MISE EN PLACE ET CONFIGURATION DU REGISTRE

3.1. CONFIGURATION DU REGISTRE POUR L'INFRASTRUCTURE AWS FOURNIE PAR L'UTILISATEUR

3.1.1. Configuration d'un secret pour l'opérateur du registre d'images

Outre les ressources **configs.imageregistry.operator.openshift.io** et ConfigMap, la configuration est fournie à l'opérateur par une ressource secrète distincte située dans l'espace de noms **openshift-image-registry**.

Le secret **image-registry-private-configuration-user** fournit les informations d'identification nécessaires à l'accès au stockage et à sa gestion. Il remplace les informations d'identification par défaut utilisées par l'opérateur, si des informations d'identification par défaut ont été trouvées.

Pour le stockage S3 sur AWS, le secret doit contenir deux clés :

- **REGISTRY_STORAGE_S3_ACCESSKEY**
- **REGISTRY_STORAGE_S3_SECRETKEY**

Procédure

- Créez un secret OpenShift Container Platform qui contient les clés requises.

```
$ oc create secret generic image-registry-private-configuration-user --from-literal=REGISTRY_STORAGE_S3_ACCESSKEY=myaccesskey --from-literal=REGISTRY_STORAGE_S3_SECRETKEY=mysecretkey --namespace openshift-image-registry
```

3.1.2. Configuration du stockage de registres pour AWS avec une infrastructure fournie par l'utilisateur

Lors de l'installation, vos informations d'identification sont suffisantes pour créer un godet Amazon S3 et l'opérateur de registre configurera automatiquement le stockage.

Si l'opérateur de registre ne peut pas créer un seau S3 et configurer automatiquement le stockage, vous pouvez créer un seau S3 et configurer le stockage à l'aide de la procédure suivante.

Conditions préalables

- Vous disposez d'un cluster sur AWS avec une infrastructure fournie par l'utilisateur.
- Pour le stockage Amazon S3, le secret doit contenir deux clés :
  - **REGISTRY_STORAGE_S3_ACCESSKEY**
  - **REGISTRY_STORAGE_S3_SECRETKEY**

Procédure

Utilisez la procédure suivante si l'opérateur de registre ne peut pas créer un seau S3 et configurer automatiquement le stockage.

1. Configurez une politique de cycle de vie des bacs pour interrompre les téléchargements multipartites incomplets datant d'un jour.

2. Complétez la configuration du stockage à l'adresse **configs.imageregistry.operator.openshift.io/cluster**:

```
$ oc edit configs.imageregistry.operator.openshift.io/cluster
```

Exemple de configuration

```
storage:
  s3:
    bucket: <bucket-name>
    region: <region-name>
```

AVERTISSEMENT

Pour sécuriser vos images de registre dans AWS, bloquez l'accès public au seau S3.

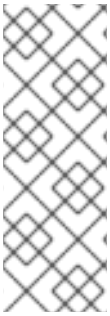
3.1.3. Paramètres de configuration de l'opérateur de registre d'images pour AWS S3

Les paramètres de configuration suivants sont disponibles pour le stockage de registre AWS S3.

ImageRegistryConfigStorageS3 contient les informations nécessaires pour configurer le registre afin qu'il utilise le service AWS S3 pour le stockage en arrière-plan. Voir la documentation du pilote de stockage S3 pour plus d'informations.

Paramètres	Description
bucket	Bucket est le nom du seau dans lequel vous souhaitez stocker les données du registre. Il est facultatif et est généré s'il n'est pas fourni.
region	Region est la région AWS dans laquelle votre bucket existe. Elle est facultative et est définie en fonction de la région AWS installée.
regionEndpoint	RegionEndpoint est le point de terminaison des services de stockage compatibles S3. Il est facultatif et les valeurs par défaut sont basées sur la région fournie.
virtualHostedStyle	VirtualHostedStyle permet d'utiliser des chemins d'accès à des seaux de style hébergé virtuel S3 avec un RegionEndpoint personnalisé. Cette option est facultative et sa valeur par défaut est false. Définissez ce paramètre pour déployer OpenShift Container Platform dans des régions cachées.

Paramètres	Description
encrypt	Encrypt spécifie si le registre stocke ou non l'image au format crypté. Il s'agit d'une option et la valeur par défaut est false.
keyID	KeyID est l'identifiant de la clé KMS à utiliser pour le cryptage. Il est facultatif. Encrypt doit être vrai, sinon ce paramètre est ignoré.
ImageRegistryConfigStorageS3CloudFront	CloudFront configure Amazon Cloudfront en tant qu'intergiciel de stockage dans un registre. Cette option est facultative.



NOTE

Lorsque la valeur du paramètre **regionEndpoint** est configurée sur une URL d'une Rados Gateway, un port explicite ne doit pas être spécifié. Par exemple, un port explicite ne doit pas être spécifié :

```
regionEndpoint: http://rook-ceph-rgw-ocs-storagecluster-cephobjectstore.openshift-storage.svc.cluster.local
```

3.2. CONFIGURATION DU REGISTRE POUR L'INFRASTRUCTURE FOURNIE PAR L'UTILISATEUR GCP

3.2.1. Configuration d'un secret pour l'opérateur du registre d'images

Outre les ressources **configs.imageregistry.operator.openshift.io** et ConfigMap, la configuration est fournie à l'opérateur par une ressource secrète distincte située dans l'espace de noms **openshift-image-registry**.

Le secret **image-registry-private-configuration-user** fournit les informations d'identification nécessaires à l'accès au stockage et à sa gestion. Il remplace les informations d'identification par défaut utilisées par l'opérateur, si des informations d'identification par défaut ont été trouvées.

Pour les GCS sur le stockage GCP, le secret est censé contenir une clé dont la valeur est le contenu d'un fichier d'informations d'identification fourni par GCP :

- **REGISTRY_STORAGE_GCS_KEYFILE**

Procédure

- Créez un secret OpenShift Container Platform qui contient les clés requises.

```
oc create secret generic image-registry-private-configuration-user --from-file=REGISTRY_STORAGE_GCS_KEYFILE=<path_to_keyfile> --namespace openshift-image-registry
```

3.2.2. Stockage de registre pour GCP avec infrastructure fournie par l'utilisateur

Vous devez configurer le support de stockage manuellement et définir les paramètres dans le registre des ressources personnalisées (CR).

Conditions préalables

- Un cluster sur GCP avec une infrastructure fournie par l'utilisateur.
- Pour configurer le stockage de registre pour GCP, vous devez fournir les informations d'identification de l'opérateur de registre.
- Pour les GCS sur le stockage GCP, le secret est censé contenir une clé dont la valeur est le contenu d'un fichier d'informations d'identification fourni par GCP :
  - **REGISTRY_STORAGE_GCS_KEYFILE**

3.2.3. Paramètres de configuration de l'opérateur du registre d'images pour le GCP GCS

Procédure

Les paramètres de configuration suivants sont disponibles pour le stockage du registre GCP GCS.

Paramètres	Description
bucket	Bucket est le nom du seau dans lequel vous souhaitez stocker les données du registre. Il est facultatif et est généré s'il n'est pas fourni.
region	La région est l'emplacement GCS dans lequel votre seau existe. Elle est facultative et est définie en fonction de la région GCS installée.
projectID	ProjectID est l'identifiant du projet GCP auquel ce bucket doit être associé. Il est facultatif.
keyID	KeyID est l'identifiant de la clé KMS à utiliser pour le chiffrement. Il est optionnel car les buckets sont cryptés par défaut sur GCP. Cela permet d'utiliser une clé de cryptage personnalisée.

3.3. CONFIGURATION DU REGISTRE POUR L'INFRASTRUCTURE OPENSTACK FOURNIE PAR L'UTILISATEUR

Vous pouvez configurer le registre d'un cluster qui s'exécute sur votre propre infrastructure Red Hat OpenStack Platform (RHOSP).

3.3.1. Configuration de l'opérateur de registre d'images pour qu'il fasse confiance au stockage Swift

Vous devez configurer l'opérateur de registre d'images pour qu'il fasse confiance au stockage Swift de Red Hat OpenStack Platform (RHOSP).

Procédure

- À partir d'une ligne de commande, entrez la commande suivante pour modifier la valeur du champ **spec.disableRedirect** de l'objet **config.imageregistry** en **true**:

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":
{"disableRedirect":true}}'
```

3.3.2. Configuration d'un secret pour l'opérateur du registre d'images

Outre les ressources **configs.imageregistry.operator.openshift.io** et ConfigMap, la configuration est fournie à l'opérateur par une ressource secrète distincte située dans l'espace de noms **openshift-image-registry**.

Le secret **image-registry-private-configuration-user** fournit les informations d'identification nécessaires à l'accès au stockage et à sa gestion. Il remplace les informations d'identification par défaut utilisées par l'opérateur, si des informations d'identification par défaut ont été trouvées.

Pour Swift sur le stockage de Red Hat OpenStack Platform (RHOSP), le secret doit contenir les deux clés suivantes :

- **REGISTRY_STORAGE_SWIFT_USER**
- **REGISTRY_STORAGE_SWIFT_PASSWORD**

Procédure

- Créez un secret OpenShift Container Platform qui contient les clés requises.

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=REGISTRY_STORAGE_SWIFT_USER=<username> --from-
literal=REGISTRY_STORAGE_SWIFT_PASSWORD=<password> -n openshift-image-
registry
```

3.3.3. Stockage du registre pour RHOSP avec une infrastructure fournie par l'utilisateur

Vous devez configurer le support de stockage manuellement et définir les paramètres dans le registre des ressources personnalisées (CR).

Conditions préalables

- Un cluster sur Red Hat OpenStack Platform (RHOSP) avec une infrastructure fournie par l'utilisateur.
- Pour configurer le stockage du registre pour RHOSP, vous devez fournir les informations d'identification de l'opérateur de registre.
- Pour Swift sur le stockage RHOSP, le secret doit contenir les deux clés suivantes :
  - **REGISTRY_STORAGE_SWIFT_USER**
  - **REGISTRY_STORAGE_SWIFT_PASSWORD**

3.3.4. Paramètres de configuration de l'opérateur de registre d'images pour RHOSP Swift

Les paramètres de configuration suivants sont disponibles pour le stockage du registre Swift de Red Hat OpenStack Platform (RHOSP).

Paramètres	Description
authURL	Cette valeur est facultative.
authVersion	Cette valeur est facultative.
container	Cette valeur est facultative.
domain	Cette valeur est facultative.
domainID	Cette valeur est facultative.
tenant	Cette valeur est facultative.
tenantID	Cette valeur est facultative.
regionName	Cette valeur est facultative.

3.4. CONFIGURATION DU REGISTRE POUR L'INFRASTRUCTURE AZURE FOURNIE PAR L'UTILISATEUR

3.4.1. Configuration d'un secret pour l'opérateur du registre d'images

Outre les ressources **configs.imageregistry.operator.openshift.io** et ConfigMap, la configuration est fournie à l'opérateur par une ressource secrète distincte située dans l'espace de noms **openshift-image-registry**.

Le secret **image-registry-private-configuration-user** fournit les informations d'identification nécessaires à l'accès au stockage et à sa gestion. Il remplace les informations d'identification par défaut utilisées par l'opérateur, si des informations d'identification par défaut ont été trouvées.

Pour le stockage de registre Azure, le secret doit contenir une clé dont la valeur est le contenu d'un fichier d'informations d'identification fourni par Azure :

- **REGISTRY_STORAGE_AZURE_ACCOUNTKEY**

Procédure

- Créez un secret OpenShift Container Platform qui contient la clé requise.

```
$ oc create secret generic image-registry-private-configuration-user --from-literal=REGISTRY_STORAGE_AZURE_ACCOUNTKEY=<accountkey> --namespace openshift-image-registry
```

3.4.2. Configuration du stockage de registre pour Azure

Lors de l'installation, vos informations d'identification sont suffisantes pour créer Azure Blob Storage, et l'opérateur de registre configure automatiquement le stockage.

Conditions préalables

- Un cluster sur Azure avec une infrastructure fournie par l'utilisateur.
- Pour configurer le stockage de registre pour Azure, fournissez les informations d'identification de l'opérateur de registre.
- Pour le stockage Azure, le secret doit contenir une clé :
  - **REGISTRY_STORAGE_AZURE_ACCOUNTKEY**

Procédure

1. Créer un conteneur de stockage Azure.
2. Complétez la configuration du stockage à l'adresse **configs.imageregistry.operator.openshift.io/cluster**:

```
$ oc edit configs.imageregistry.operator.openshift.io/cluster
```

Exemple de configuration

```
storage:
  azure:
    accountName: <storage-account-name>
    container: <container-name>
```

3.4.3. Configuration du stockage de registre pour Azure Government

Lors de l'installation, vos informations d'identification sont suffisantes pour créer Azure Blob Storage, et l'opérateur de registre configure automatiquement le stockage.

Conditions préalables

- Un cluster sur Azure avec une infrastructure fournie par l'utilisateur dans une région gouvernementale.
- Pour configurer le stockage de registre pour Azure, fournissez les informations d'identification de l'opérateur de registre.
- Pour le stockage Azure, le secret doit contenir une clé :
  - **REGISTRY_STORAGE_AZURE_ACCOUNTKEY**

Procédure

1. Créer un conteneur de stockage Azure.
2. Complétez la configuration du stockage à l'adresse **configs.imageregistry.operator.openshift.io/cluster**:

```
$ oc edit configs.imageregistry.operator.openshift.io/cluster
```

Exemple de configuration

-

```
storage:
  azure:
    accountName: <storage-account-name>
    container: <container-name>
    cloudName: AzureUSGovernmentCloud (1)
```

(1) **cloudName** est le nom de l'environnement cloud Azure, qui peut être utilisé pour configurer le SDK Azure avec les points d'extrémité Azure API appropriés. La valeur par défaut est **AzurePublicCloud**. Vous pouvez également définir **cloudName** sur **AzureUSGovernmentCloud**, **AzureChinaCloud** ou **AzureGermanCloud** avec des informations d'identification suffisantes.

3.5. CONFIGURATION DU REGISTRE POUR RHOSP

3.5.1. Configurer un registre d'images avec un stockage personnalisé sur des clusters fonctionnant sous RHOSP

Après avoir installé un cluster sur Red Hat OpenStack Platform (RHOSP), vous pouvez utiliser un volume Cinder qui se trouve dans une zone de disponibilité spécifique pour le stockage du registre.

Procédure

1. Créez un fichier YAML qui spécifie la classe de stockage et la zone de disponibilité à utiliser. Par exemple :

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: custom-csi-storageclass
provisioner: cinder.csi.openstack.org
volumeBindingMode: WaitForFirstConsumer
allowVolumeExpansion: true
parameters:
  availability: <availability_zone_name>
```

NOTE

OpenShift Container Platform ne vérifie pas l'existence de la zone de disponibilité que vous choisissez. Vérifiez le nom de la zone de disponibilité avant d'appliquer la configuration.

2. Appliquer la configuration à partir d'une ligne de commande :

```
oc apply -f <storage_class_file_name>
```

Exemple de sortie

```
storageclass.storage.k8s.io/custom-csi-storageclass created
```

3. Créez un fichier YAML qui spécifie une revendication de volume persistant (PVC) utilisant votre classe de stockage et l'espace de noms **openshift-image-registry**. Par exemple :

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: csi-pvc-imageregistry
  namespace: openshift-image-registry (1)
  annotations:
    imageregistry.openshift.io: "true"
spec:
  accessModes:
  - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 100Gi (2)
  storageClassName: <your_custom_storage_class> (3)
```

1. Saisissez l'espace de noms **openshift-image-registry**. Cet espace de noms permet à l'opérateur du registre des images de cluster de consommer le PVC.
2. En option : Ajustez la taille du volume.
3. Saisissez le nom de la classe de stockage que vous avez créée.

4. Appliquer la configuration à partir d'une ligne de commande :

```
oc apply -f <nom_du_fichier_de_vc>
```

Exemple de sortie

```
persistentvolumeclaim/csi-pvc-imageregistry created
```

5. Remplacer la revendication originale du volume persistant dans la configuration du registre d'images par la nouvelle revendication :

```
$ oc patch configs.imageregistry.operator.openshift.io/cluster --type 'json' -p='[{"op":
"replace", "path": "/spec/storage/pvc/claim", "value": "csi-pvc-imageregistry"}]'
```

Exemple de sortie

```
config.imageregistry.operator.openshift.io/cluster patched
```

Au cours des minutes suivantes, la configuration est mise à jour.

Vérification

Pour confirmer que le registre utilise les ressources que vous avez définies :

1. Vérifiez que la valeur de la réclamation PVC est identique au nom que vous avez fourni dans votre définition du PVC :

```
$ oc get configs.imageregistry.operator.openshift.io/cluster -o yaml
```

Exemple de sortie

```
...
status:
  ...
  managementState: Managed
  pvc:
    claim: csi-pvc-imageregistry
...
```

2. Vérifiez que le statut du PVC est **Bound**:

```
$ oc get pvc -n openshift-image-registry csi-pvc-imageregistry
```

Exemple de sortie

```
NAME                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS             AGE
csi-pvc-imageregistry    Bound    pvc-72a8f9c9-f462-11e8-b6b6-fa163e18b7b5   100Gi
RWO            custom-csi-storageclass   11m
```

3.6. CONFIGURATION DU REGISTRE POUR LE MÉTAL NU

3.6.1. Registre d'images supprimé lors de l'installation

Sur les plateformes qui ne fournissent pas de stockage d'objets partageables, l'opérateur de registre d'images OpenShift s'amorce lui-même en tant que **Removed**. Cela permet à **openshift-installer** de réaliser des installations sur ces types de plateformes.

Après l'installation, vous devez modifier la configuration de l'opérateur du registre des images pour faire passer le site **managementState** de **Removed** à **Managed**.

NOTE

La console Prometheus fournit une alerte **ImageRegistryRemoved**, par exemple :

"Image Registry has been removed. **ImageStreamTags** il se peut que les fichiers **BuildConfigs** et **DeploymentConfigs** qui font référence à **ImageStreamTags** ne fonctionnent pas comme prévu. Veuillez configurer le stockage et mettre à jour la configuration à l'état **Managed** en éditant configs.imageregistry.operator.openshift.io."

3.6.2. Modification de l'état de gestion du registre d'images

Pour démarrer le registre d'images, vous devez modifier la configuration de l'opérateur du registre d'images **managementState** de **Removed** à **Managed**.

Procédure

- Modifier la configuration de l'opérateur du registre d'images **managementState** de **Removed** à **Managed**. Par exemple :

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":
{"managementState":"Managed"}}'
```

3.6.3. Configuration du stockage du registre d'images

L'opérateur de registre d'images n'est pas disponible initialement pour les plates-formes qui ne fournissent pas de stockage par défaut. Après l'installation, vous devez configurer votre registre pour utiliser le stockage afin que l'opérateur de registre soit disponible.

Des instructions sont données pour la configuration d'un volume persistant, qui est nécessaire pour les clusters de production. Le cas échéant, des instructions sont fournies pour configurer un répertoire vide comme emplacement de stockage, ce qui n'est possible que pour les clusters de non-production.

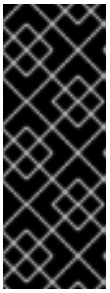
Des instructions supplémentaires sont fournies pour permettre au registre d'images d'utiliser des types de stockage en bloc en utilisant la stratégie de déploiement **Recreate** lors des mises à niveau.

3.6.3.1. Configuration du stockage du registre pour les installations "bare metal" et autres installations manuelles

En tant qu'administrateur de cluster, vous devez, après l'installation, configurer votre registre pour utiliser le stockage.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez un cluster qui utilise des nœuds Red Hat Enterprise Linux CoreOS (RHCOS) approvisionnés manuellement, tels que du métal nu.
- Vous avez provisionné un stockage persistant pour votre cluster, tel que Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform prend en charge l'accès **ReadWriteOnce** pour le stockage du registre d'images lorsque vous n'avez qu'une seule réplique. L'accès **ReadWriteOnce** nécessite également que le registre utilise la stratégie de déploiement **Recreate**. Pour déployer un registre d'images qui prend en charge la haute disponibilité avec deux répliques ou plus, l'accès **ReadWriteMany** est requis.

- Doit avoir une capacité de 100Gi.

Procédure

1. Pour configurer votre registre afin qu'il utilise le stockage, modifiez l'adresse **spec.storage.pvc** dans la ressource **configs.imageregistry/cluster**.

NOTE

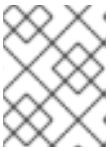
Lorsque vous utilisez un espace de stockage partagé, vérifiez vos paramètres de sécurité afin d'empêcher tout accès extérieur.

2. Vérifiez que vous n'avez pas de pod de registre :

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Exemple de sortie

```
No resources found in openshift-image-registry namespace
```



NOTE

Si vous avez un pod de registre dans votre sortie, il n'est pas nécessaire de poursuivre cette procédure.

3. Vérifier la configuration du registre :

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Exemple de sortie

```
storage:
  pvc:
    claim:
```

Laissez le champ **claim** vide pour permettre la création automatique d'un PVC **image-registry-storage**.

4. Vérifier l'état de **clusteroperator**:

```
$ oc get clusteroperator image-registry
```

Exemple de sortie

```
NAME             VERSION       AVAILABLE   PROGRESSING   DEGRADED   SINCE
MESSAGE
image-registry   4.12          True        False         False      6h50m
```

5. Veillez à ce que votre registre soit géré pour permettre la création et l'envoi d'images.

- Exécutez :

```
$ oc edit configs.imageregistry/cluster
```

Modifiez ensuite la ligne

```
managementState: Removed
```

à

```
managementState: Managed
```

3.6.3.2. Configuration du stockage pour le registre d'images dans les clusters de non-production

Vous devez configurer le stockage pour l'opérateur de registre d'images. Pour les clusters de non-production, vous pouvez définir le registre d'images dans un répertoire vide. Dans ce cas, toutes les images seront perdues si vous redémarrez le registre.

Procédure

- Pour définir le stockage du registre d'images dans un répertoire vide :

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":{"storage":{"emptyDir":{}}}}'
```

AVERTISSEMENT

Configurez cette option pour les clusters de non-production uniquement.

Si vous exécutez cette commande avant que l'opérateur de registre d'images n'initialise ses composants, la commande **oc patch** échoue avec l'erreur suivante :

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

Attendez quelques minutes et exécutez à nouveau la commande.

3.6.3.3. Configuration du stockage dans le registre des blocs

Pour permettre au registre d'images d'utiliser des types de stockage en bloc lors des mises à niveau en tant qu'administrateur de cluster, vous pouvez utiliser la stratégie de déploiement **Recreate**.

IMPORTANT

Les volumes de stockage en bloc sont pris en charge mais ne sont pas recommandés pour une utilisation avec le registre d'images sur des clusters de production. Une installation où le registre est configuré sur un stockage en bloc n'est pas hautement disponible car le registre ne peut pas avoir plus d'une réplique.

Procédure

1. Pour définir le stockage du registre d'images comme un type de stockage en bloc, corrigez le registre afin qu'il utilise la stratégie de déploiement **Recreate** et s'exécute avec une seule réplique (**1**) :

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec":{"rolloutStrategy":"Recreate","replicas":1}}'
```

2. Provisionnez le PV pour le périphérique de stockage en bloc et créez un PVC pour ce volume. Le volume bloc demandé utilise le mode d'accès ReadWriteOnce (RWO).

3. Modifiez la configuration du registre de manière à ce qu'elle fasse référence au PVC correct.

3.6.3.4. Configurer l'opérateur de registre d'images pour utiliser le stockage Ceph RGW avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Ceph RGW.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Ceph RGW.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **ocs-storagecluster-ceph-rgw**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: rgwtest
  namespace: openshift-storage
spec:
  storageClassName: ocs-storagecluster-ceph-rgw
  generateBucketName: rgwtest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage rgwtest -o jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage rgwtest -o yaml | grep -w
"AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage rgwtest -o yaml | grep
-w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --namespace openshift-image-registry
```

5. Créez une route de cryptage pour Ceph RGW en entrant la commande suivante :

```
$ oc create route reencrypt <route_name> --service=rook-ceph-rgw-ocs-storagecluster-cephobjectstore --port=https -n openshift-storage
```

   a. Obtenez l'hôte de la route en entrant la commande suivante :

```
route_host=$(oc get route <route_name> -n openshift-storage -o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Ceph RGW en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":{"managementState":"Managed","replicas":2,"storage":{"managementState":"Unmanaged","s3":{"bucket":"\"${bucket_name}\"","region":"us-east-1","regionEndpoint":"\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedCA":{"name":"image-registry-s3-bundle"}}}}}' --type=merge
```

3.6.3.5. Configurer l'opérateur de registre d'images pour utiliser le stockage Noobaa avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Noobaa.

Conditions préalables

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

- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Noobaa.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **openshift-storage.noobaa.io**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: noobaatest
  namespace: openshift-storage
spec:
  storageClassName: openshift-storage.noobaa.io
  generateBucketName: noobaatest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage noobaatest -o
jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage noobaatest -o yaml | grep -w
"AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage noobaatest -o yaml |
grep -w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-
literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --
namespace openshift-image-registry
```

5. Obtenez l'hôte de la route en entrant la commande suivante :

```
$ route_host=$(oc get route s3 -n openshift-storage -o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n
openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Nooba en entrant la commande suivante :

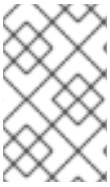
```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","s3":{"bucket":"\"${bucket_name}\"","region":"us-east-
1","regionEndpoint":"\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedC
A":{"name":"image-registry-s3-bundle"}}}}}' --type=merge
```

3.6.4. Configurer l'opérateur de registre d'images pour utiliser le stockage CephFS avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage CephFS.



NOTE

CephFS utilise le stockage par revendication de volume persistant (PVC). Il n'est pas recommandé d'utiliser les PVC pour le stockage des registres d'images s'il existe d'autres options, telles que Ceph RGW ou Noobaa.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé OpenShift Data Foundation Operator pour fournir le stockage d'objets et le stockage de fichiers CephFS.

Procédure

1. Créez un PVC pour utiliser la classe de stockage **cephfs**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: registry-storage-pvc
 namespace: openshift-image-registry
spec:
 accessModes:
```

```
 - ReadWriteMany
 resources:
   requests:
     storage: 100Gi
 storageClassName: ocs-storagecluster-cephfs
EOF
```

2. Configurez le registre d'images pour utiliser le système de stockage de fichiers CephFS en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","pvc":{"claim":"registry-storage-pvc"}}}}' --type=merge
```

3.6.5. Ressources complémentaires

- Technologie de stockage configurable recommandée
- Configurer Image Registry pour utiliser OpenShift Data Foundation

3.7. CONFIGURATION DU REGISTRE POUR VSPHERE

3.7.1. Registre d'images supprimé lors de l'installation

Sur les plateformes qui ne fournissent pas de stockage d'objets partageables, l'opérateur de registre d'images OpenShift s'amorce lui-même en tant que **Removed**. Cela permet à **openshift-installer** de réaliser des installations sur ces types de plateformes.

Après l'installation, vous devez modifier la configuration de l'opérateur du registre des images pour faire passer le site **managementState** de **Removed** à **Managed**.

NOTE

La console Prometheus fournit une alerte **ImageRegistryRemoved**, par exemple :

"Image Registry has been removed. **ImageStreamTags** il se peut que les fichiers **BuildConfigs** et **DeploymentConfigs** qui font référence à **ImageStreamTags** ne fonctionnent pas comme prévu. Veuillez configurer le stockage et mettre à jour la configuration à l'état **Managed** en éditant configs.imageregistry.operator.openshift.io."

3.7.2. Modification de l'état de gestion du registre d'images

Pour démarrer le registre d'images, vous devez modifier la configuration de l'opérateur du registre d'images **managementState** de **Removed** à **Managed**.

Procédure

- Modifier la configuration de l'opérateur du registre d'images **managementState** de **Removed** à **Managed**. Par exemple :

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":
{"managementState":"Managed"}}'
```

3.7.3. Configuration du stockage du registre d'images

L'opérateur de registre d'images n'est pas disponible initialement pour les plates-formes qui ne fournissent pas de stockage par défaut. Après l'installation, vous devez configurer votre registre pour utiliser le stockage afin que l'opérateur de registre soit disponible.

Des instructions sont données pour la configuration d'un volume persistant, qui est nécessaire pour les clusters de production. Le cas échéant, des instructions sont fournies pour configurer un répertoire vide comme emplacement de stockage, ce qui n'est possible que pour les clusters de non-production.

Des instructions supplémentaires sont fournies pour permettre au registre d'images d'utiliser des types de stockage en bloc en utilisant la stratégie de déploiement **Recreate** lors des mises à niveau.

3.7.3.1. Configuration du stockage de registre pour VMware vSphere

En tant qu'administrateur de cluster, vous devez, après l'installation, configurer votre registre pour utiliser le stockage.

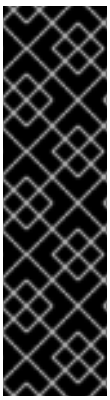
Conditions préalables

- Permissions de l'administrateur du cluster.
- Un cluster sur VMware vSphere.
- Stockage persistant provisionné pour votre cluster, tel que Red Hat OpenShift Data Foundation.

IMPORTANT

OpenShift Container Platform prend en charge l'accès **ReadWriteOnce** pour le stockage du registre d'images lorsque vous n'avez qu'une seule réplique. L'accès **ReadWriteOnce** nécessite également que le registre utilise la stratégie de déploiement **Recreate**. Pour déployer un registre d'images qui prend en charge la haute disponibilité avec deux répliques ou plus, l'accès **ReadWriteMany** est requis.

- Doit avoir une capacité de "100Gi".



IMPORTANT

Les tests montrent des problèmes avec l'utilisation du serveur NFS sur RHEL comme backend de stockage pour les services principaux. Cela inclut OpenShift Container Registry et Quay, Prometheus pour la surveillance du stockage, et Elasticsearch pour la journalisation du stockage. Par conséquent, il n'est pas recommandé d'utiliser le serveur NFS de RHEL pour sauvegarder les PV utilisés par les services principaux.

D'autres implémentations NFS sur le marché peuvent ne pas avoir ces problèmes. Contactez le vendeur de l'implémentation NFS pour plus d'informations sur les tests qui ont pu être réalisés avec ces composants de base d'OpenShift Container Platform.

Procédure

1. Pour configurer votre registre afin qu'il utilise le stockage, modifiez l'adresse **spec.storage.pvc** dans la ressource **configs.imageregistry/cluster**.

NOTE

Lorsque vous utilisez un espace de stockage partagé, vérifiez vos paramètres de sécurité afin d'empêcher tout accès extérieur.

2. Vérifiez que vous n'avez pas de pod de registre :

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Exemple de sortie

```
No resourses found in openshift-image-registry namespace
```

NOTE

Si vous avez un pod de registre dans votre sortie, il n'est pas nécessaire de poursuivre cette procédure.

3. Vérifier la configuration du registre :

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Exemple de sortie

```
storage:
  pvc:
    claim: 1
```

1 Laissez le champ **claim** vide pour permettre la création automatique d'une réclamation de volume persistant (PVC) **image-registry-storage**. Le PVC est généré en fonction de la classe de stockage par défaut. Cependant, il faut savoir que la classe de stockage par défaut peut fournir des volumes ReadWriteOnce (RWO), tels qu'un RADOS Block Device (RBD), ce qui peut poser des problèmes lors de la réplication sur plusieurs répliques.

4. Vérifier l'état de **clusteroperator**:

```
$ oc get clusteroperator image-registry
```

Exemple de sortie

```
NAME             VERSION                             AVAILABLE   PROGRESSING   DEGRADED
SINCE   MESSAGE
image-registry   4.7                                 True        False         False     6h50m
```

3.7.3.2. Configuration du stockage pour le registre d'images dans les clusters de non-production

Vous devez configurer le stockage pour l'opérateur de registre d'images. Pour les clusters de non-production, vous pouvez définir le registre d'images dans un répertoire vide. Dans ce cas, toutes les images seront perdues si vous redémarrez le registre.

Procédure

- Pour définir le stockage du registre d'images dans un répertoire vide :

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":
{"storage":{"emptyDir":{}}}}'
```

AVERTISSEMENT

Configurez cette option pour les clusters de non-production uniquement.

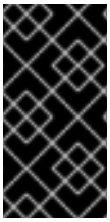
Si vous exécutez cette commande avant que l'opérateur de registre d'images n'initialise ses composants, la commande **oc patch** échoue avec l'erreur suivante :

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

Attendez quelques minutes et exécutez à nouveau la commande.

3.7.3.3. Configuration du stockage dans le registre de blocs pour VMware vSphere

Pour permettre au registre d'images d'utiliser des types de stockage en bloc tels que vSphere Virtual Machine Disk (VMDK) lors des mises à niveau en tant qu'administrateur de cluster, vous pouvez utiliser la stratégie de déploiement **Recreate**.



IMPORTANT

Les volumes de stockage en bloc sont pris en charge mais ne sont pas recommandés pour une utilisation avec le registre d'images sur des clusters de production. Une installation où le registre est configuré sur un stockage en bloc n'est pas hautement disponible car le registre ne peut pas avoir plus d'une réplique.

Procédure

1. Pour définir le stockage du registre d'images comme un type de stockage en bloc, corrigez le registre afin qu'il utilise la stratégie de déploiement **Recreate** et qu'il s'exécute avec la seule réplique **1**:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec":
{"rolloutStrategy":"Recreate","replicas":1}}'
```

2. Provisionnez le PV pour le périphérique de stockage en bloc et créez un PVC pour ce volume. Le volume bloc demandé utilise le mode d'accès ReadWriteOnce (RWO).

   a. Créez un fichier **pvc.yaml** avec le contenu suivant pour définir un objet VMware vSphere **PersistentVolumeClaim**:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
```

```
    name: image-registry-storage 1
    namespace: openshift-image-registry 2
  spec:
    accessModes:
    - ReadWriteOnce 3
    resources:
      requests:
        storage: 100Gi 4
```

1. Un nom unique qui représente l'objet **PersistentVolumeClaim**.
2. L'espace de noms de l'objet **PersistentVolumeClaim**, qui est **openshift-image-registry**.
3. Le mode d'accès de la demande de volume persistant. Avec **ReadWriteOnce**, le volume peut être monté avec des autorisations de lecture et d'écriture par un seul nœud.
4. Taille de la demande de volume persistant.

b. Créer l'objet **PersistentVolumeClaim** à partir du fichier :

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Modifiez la configuration du registre de manière à ce qu'elle fasse référence au PVC correct :

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Exemple de sortie

```
storage:
  pvc:
    claim: 1
```

1. La création d'un PVC personnalisé vous permet de laisser le champ **claim** vide pour la création automatique par défaut d'un PVC **image-registry-storage**.

Pour obtenir des instructions sur la configuration du stockage du registre afin qu'il référence le PVC correct, voir Configuration du registre pour vSphere .

3.7.3.4. Configurer l'opérateur de registre d'images pour utiliser le stockage Ceph RGW avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Ceph RGW.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Ceph RGW.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **ocs-storagecluster-ceph-rgw**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: rgwtest
  namespace: openshift-storage
spec:
  storageClassName: ocs-storagecluster-ceph-rgw
  generateBucketName: rgwtest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage rgwtest -o jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage rgwtest -o yaml | grep -w
"AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage rgwtest -o yaml | grep
-w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-
literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --
namespace openshift-image-registry
```

5. Créez une route de cryptage pour Ceph RGW en entrant la commande suivante :

```
$ oc create route reencrypt <route_name> --service=rook-ceph-rgw-ocs-storagecluster-
cephobjectstore --port=https -n openshift-storage
```

   a. Obtenez l'hôte de la route en entrant la commande suivante :

```
route_host=$(oc get route <route_name> -n openshift-storage -
o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n
openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Ceph RGW en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","s3":{"bucket":\"${bucket_name}\"","region":"us-east-
1","regionEndpoint":\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedC
A":{"name":"image-registry-s3-bundle"}}}}}' --type=merge
```

3.7.3.5. Configurer l'opérateur de registre d'images pour utiliser le stockage Noobaa avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Noobaa.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Noobaa.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **openshift-storage.noobaa.io**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: noobaatest
```

```
  namespace: openshift-storage
spec:
  storageClassName: openshift-storage.noobaa.io
  generateBucketName: noobaatest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage noobaatest -o
jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage noobaatest -o yaml | grep -w
"AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage noobaatest -o yaml |
grep -w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-
literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --
namespace openshift-image-registry
```

5. Obtenez l'hôte de la route en entrant la commande suivante :

```
$ route_host=$(oc get route s3 -n openshift-storage -o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n
openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Nooba en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","s3":{"bucket":"\"${bucket_name}\"","region":"us-east-
1","regionEndpoint":"\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedC
A":{"name":"image-registry-s3-bundle"}}}}}' --type=merge
```

3.7.4. Configurer l'opérateur de registre d'images pour utiliser le stockage CephFS avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage CephFS.

NOTE

CephFS utilise le stockage par revendication de volume persistant (PVC). Il n'est pas recommandé d'utiliser les PVC pour le stockage des registres d'images s'il existe d'autres options, telles que Ceph RGW ou Noobaa.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé OpenShift Data Foundation Operator pour fournir le stockage d'objets et le stockage de fichiers CephFS.

Procédure

1. Créez un PVC pour utiliser la classe de stockage **cephfs**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: registry-storage-pvc
 namespace: openshift-image-registry
spec:
 accessModes:
 - ReadWriteMany
 resources:
   requests:
     storage: 100Gi
 storageClassName: ocs-storagecluster-cephfs
EOF
```

2. Configurez le registre d'images pour utiliser le système de stockage de fichiers CephFS en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","pvc":{"claim":"registry-storage-pvc"}}}}' --type=merge
```

3.7.5. Ressources complémentaires

- Technologie de stockage configurable recommandée
- Configurer Image Registry pour utiliser OpenShift Data Foundation

3.8. CONFIGURATION DU REGISTRE POUR RED HAT OPENSHIFT DATA FOUNDATION

Pour configurer le registre d'images interne sur bare metal et vSphere afin d'utiliser le stockage Red Hat OpenShift Data Foundation, vous devez installer OpenShift Data Foundation et ensuite configurer le registre d'images à l'aide de Ceph ou Noobaa.

3.8.1. Configurer l'opérateur de registre d'images pour utiliser le stockage Ceph RGW avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Ceph RGW.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Ceph RGW.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **ocs-storagecluster-ceph-rgw**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: rgwtest
  namespace: openshift-storage
spec:
  storageClassName: ocs-storagecluster-ceph-rgw
  generateBucketName: rgwtest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage rgwtest -o jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage rgwtest -o yaml | grep -w "AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage rgwtest -o yaml | grep -w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --namespace openshift-image-registry
```

5. Créez une route de cryptage pour Ceph RGW en entrant la commande suivante :

```
$ oc create route reencrypt <route_name> --service=rook-ceph-rgw-ocs-storagecluster-cephobjectstore --port=https -n openshift-storage
```

   a. Obtenez l'hôte de la route en entrant la commande suivante :

```
route_host=$(oc get route <route_name> -n openshift-storage -o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Ceph RGW en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":{"managementState":"Managed","replicas":2,"storage":{"managementState":"Unmanaged","s3":{"bucket":"\"${bucket_name}\"","region":"us-east-1","regionEndpoint":"\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedCA":{"name":"image-registry-s3-bundle"}}}}' --type=merge
```

3.8.2. Configurer l'opérateur de registre d'images pour utiliser le stockage Noobaa avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site

- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage Noobaa.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé l'opérateur OpenShift Data Foundation pour fournir le stockage d'objets et le stockage d'objets Noobaa.

Procédure

1. Créez la demande de seau d'objets à l'aide de la classe de stockage **openshift-storage.noobaa.io**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: noobaatest
  namespace: openshift-storage
spec:
  storageClassName: openshift-storage.noobaa.io
  generateBucketName: noobaatest
EOF
```

2. Obtenez le nom du seau en entrant la commande suivante :

```
$ bucket_name=$(oc get obc -n openshift-storage noobaatest -o
jsonpath='{.spec.bucketName}')
```

3. Obtenez les informations d'identification AWS en entrant les commandes suivantes :

```
$ AWS_ACCESS_KEY_ID=$(oc get secret -n openshift-storage noobaatest -o yaml | grep -w
"AWS_ACCESS_KEY_ID:" | head -n1 | awk '{print $2}' | base64 --decode)
```

```
$ AWS_SECRET_ACCESS_KEY=$(oc get secret -n openshift-storage noobaatest -o yaml |
grep -w "AWS_SECRET_ACCESS_KEY:" | head -n1 | awk '{print $2}' | base64 --decode)
```

4. Créez le secret **image-registry-private-configuration-user** avec les informations d'identification AWS pour le nouveau seau sous **openshift-image-registry project** en entrant la commande suivante :

```
$ oc create secret generic image-registry-private-configuration-user --from-
literal=REGISTRY_STORAGE_S3_ACCESSKEY=${AWS_ACCESS_KEY_ID} --from-
literal=REGISTRY_STORAGE_S3_SECRETKEY=${AWS_SECRET_ACCESS_KEY} --
namespace openshift-image-registry
```

5. Obtenez l'hôte de la route en entrant la commande suivante :

```
$ route_host=$(oc get route s3 -n openshift-storage -o=jsonpath='{.spec.host}')
```

6. Créez une carte de configuration qui utilise un certificat d'entrée en entrant les commandes suivantes :

```
$ oc extract secret/router-certs-default  -n openshift-ingress  --confirm
```

```
$ oc create configmap image-registry-s3-bundle --from-file=ca-bundle.crt=./tls.crt  -n
openshift-config
```

7. Configurez le registre d'images pour utiliser le stockage d'objets Nooba en entrant la commande suivante :

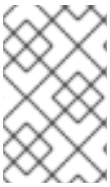
```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","s3":{"bucket":"\"${bucket_name}\"","region":"us-east-
1","regionEndpoint":"\"https://${route_host}\"","virtualHostedStyle":false,"encrypt":false,"trustedC
A":{"name":"image-registry-s3-bundle"}}}}}' --type=merge
```

3.8.3. Configurer l'opérateur de registre d'images pour utiliser le stockage CephFS avec Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation intègre plusieurs types de stockage que vous pouvez utiliser avec le registre d'images interne :

- Ceph, un système de fichiers partagé et distribué et un système de stockage d'objets sur site
- NooBaa, une passerelle d'objets multicloud

Ce document décrit la procédure à suivre pour configurer le registre d'images afin d'utiliser le stockage CephFS.



NOTE

CephFS utilise le stockage par revendication de volume persistant (PVC). Il n'est pas recommandé d'utiliser les PVC pour le stockage des registres d'images s'il existe d'autres options, telles que Ceph RGW ou Noobaa.

Conditions préalables

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.
- Vous avez accès à la console web de OpenShift Container Platform.
- Vous avez installé le CLI **oc**.
- Vous avez installé OpenShift Data Foundation Operator pour fournir le stockage d'objets et le stockage de fichiers CephFS.

Procédure

1. Créez un PVC pour utiliser la classe de stockage **cephfs**. Par exemple :

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: registry-storage-pvc
 namespace: openshift-image-registry
spec:
 accessModes:
 - ReadWriteMany
 resources:
   requests:
     storage: 100Gi
 storageClassName: ocs-storagecluster-cephfs
EOF
```

2. Configurez le registre d'images pour utiliser le système de stockage de fichiers CephFS en entrant la commande suivante :

```
$ oc patch config.image/cluster -p '{"spec":
{"managementState":"Managed","replicas":2,"storage":
{"managementState":"Unmanaged","pvc":{"claim":"registry-storage-pvc"}}}}' --type=merge
```

3.8.4. Ressources complémentaires

- Configurer Image Registry pour utiliser OpenShift Data Foundation
- Guide d'optimisation des performances pour Multicloud Object Gateway (NooBaa)

CHAPITRE 4. ACCÈS AU REGISTRE

Les sections suivantes contiennent des instructions sur l'accès au registre, y compris l'affichage des journaux et des mesures, ainsi que sur la sécurisation et l'exposition du registre.

Vous pouvez accéder directement au registre pour invoquer les commandes **podman**. Cela vous permet d'envoyer ou de retirer des images du registre intégré directement à l'aide d'opérations telles que **podman push** ou **podman pull**. Pour ce faire, vous devez être connecté au registre à l'aide de la commande **podman login**. Les opérations que vous pouvez effectuer dépendent de vos droits d'utilisateur, comme décrit dans les sections suivantes.

4.1. CONDITIONS PRÉALABLES

- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle de cluster-admin.
- Vous devez avoir configuré un fournisseur d'identité (IDP).
- Pour extraire des images, par exemple en utilisant la commande **podman pull**, l'utilisateur doit avoir le rôle **registry-viewer**. Pour ajouter ce rôle, exécutez la commande suivante :

```
$ oc policy add-role-to-user registry-viewer < nom_de_l'utilisateur>
```

- Pour écrire ou pousser des images, par exemple lors de l'utilisation de la commande **podman push**:
  - L'utilisateur doit avoir le rôle **registry-editor**. Pour ajouter ce rôle, exécutez la commande suivante :

```
$ oc policy add-role-to-user registry-editor <user_name>
```

  - Votre cluster doit avoir un projet existant dans lequel les images peuvent être poussées.

4.2. ACCÈS AU REGISTRE DIRECTEMENT À PARTIR DU CLUSTER

Vous pouvez accéder au registre depuis l'intérieur du cluster.

Procédure

Accédez au registre depuis le cluster en utilisant des routes internes :

1. Accéder au nœud en obtenant son nom :

```
$ oc get nodes
```

```
$ oc debug nodes/<node_name>
```

2. Pour permettre l'accès à des outils tels que **oc** et **podman** sur le nœud, changez votre répertoire racine en /**host**:

```
sh-4.2# chroot /host
```

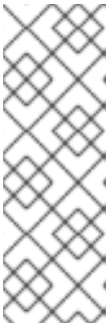
3. Connectez-vous au registre des images de conteneurs en utilisant votre jeton d'accès :

```
sh-4.2# oc login -u kubeadmin -p <mot_de_passe_du_log_d'installation> https://api-int.<cluster_name>.<base_domain>:6443
```

```
sh-4.2# podman login -u kubeadmin -p $(oc whoami -t) image-registry.openshift-image-registry.svc:5000
```

Un message de confirmation de la connexion devrait s'afficher, comme par exemple :

```
Login Succeeded!
```

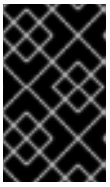


NOTE

Vous pouvez indiquer n'importe quelle valeur pour le nom d'utilisateur ; le jeton contient toutes les informations nécessaires. La transmission d'un nom d'utilisateur contenant des deux-points entraînera un échec de la connexion.

Étant donné que c'est l'opérateur du registre des images qui crée l'itinéraire, celui-ci sera probablement similaire à **default-route-openshift-image-registry.<cluster_name>**.

4. Effectuez les opérations **podman pull** et **podman push** sur votre registre :



IMPORTANT

Vous pouvez extraire des images arbitraires, mais si vous avez ajouté le rôle **system:registry**, vous ne pouvez que pousser des images vers le registre de votre projet.

Dans les exemples suivants, utilisez :

Composant	Valeur
<registry_ip>	**172.30.124.220**
<port>	**5000**
<project>	**openshift**
<image>	**image**
<tag>	omis (par défaut **latest**)

a. Tirer une image arbitraire :

```
sh-4.2# podman pull <name.io>/<image>
```

b. Marquez la nouvelle image avec la forme **<registry_ip>:<port>/<project>/<image>**. Le nom du projet doit apparaître dans cette spécification d'extraction pour que OpenShift Container Platform place correctement l'image dans le registre et y accède ultérieurement :

```
sh-4.2# podman tag <name.io>/<image> image-registry.openshift-image-
registry.svc:5000/openshift/<image>
```

NOTE

Vous devez avoir le rôle **system:image-builder** pour le projet spécifié, qui permet à l'utilisateur d'écrire ou de pousser une image. Dans le cas contraire, la commande **podman push** de l'étape suivante échouera. Pour tester, vous pouvez créer un nouveau projet pour pousser l'image.

c. Transférez l'image nouvellement étiquetée dans votre registre :

```
sh-4.2# podman push image-registry.openshift-image-
registry.svc:5000/openshift/<image>
```

4.3. VÉRIFICATION DE L'ÉTAT DES PODS DE REGISTRE

En tant qu'administrateur de cluster, vous pouvez dresser la liste des pods de registre d'images en cours d'exécution dans le projet **openshift-image-registry** et vérifier leur état.

Conditions préalables

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

Procédure

1. Liste les pods du projet **openshift-image-registry** et visualise leur statut :

```
$ oc get pods -n openshift-image-registry
```

Exemple de sortie

```
NAME READY STATUS RESTARTS AGE
cluster-image-registry-operator-764bd7f846-qqtpb 1/1 Running 0 78m
image-registry-79fb4469f6-llrln 1/1 Running 0 77m
node-ca-hjksc 1/1 Running 0 73m
node-ca-tftj6 1/1 Running 0 77m
node-ca-wb6ht 1/1 Running 0 77m
node-ca-zvt9q 1/1 Running 0 74m
```

4.4. VISUALISATION DES JOURNAUX DE REGISTRE

Vous pouvez consulter les journaux du registre à l'aide de la commande **oc logs**.

Procédure

1. Utilisez la commande **oc logs** avec les déploiements pour afficher les journaux du registre des images de conteneurs :

```
$ oc logs deployments/image-registry -n openshift-image-registry
```

Exemple de sortie

```
2015-05-01T19:48:36.300593110Z time="2015-05-01T19:48:36Z" level=info
msg="version=v2.0.0+unknown"
2015-05-01T19:48:36.303294724Z time="2015-05-01T19:48:36Z" level=info msg="redis not
configured" instance.id=9ed6c43d-23ee-453f-9a4b-031fea646002
2015-05-01T19:48:36.303422845Z time="2015-05-01T19:48:36Z" level=info msg="using
inmemory layerinfo cache" instance.id=9ed6c43d-23ee-453f-9a4b-031fea646002
2015-05-01T19:48:36.303433991Z time="2015-05-01T19:48:36Z" level=info msg="Using
OpenShift Auth handler"
2015-05-01T19:48:36.303439084Z time="2015-05-01T19:48:36Z" level=info msg="listening
on :5000" instance.id=9ed6c43d-23ee-453f-9a4b-031fea646002
```

4.5. ACCÈS AUX MÉTRIQUES DU REGISTRE

L'OpenShift Container Registry fournit un point de terminaison pour les métriques Prometheus. Prometheus est une boîte à outils autonome et open source de surveillance des systèmes et d'alerte.

Les métriques sont exposées dans le chemin d'accès ***/extensions/v2/metrics*** du point d'accès au registre.

Procédure

Vous pouvez accéder aux métriques en exécutant une requête de métriques à l'aide d'un rôle de cluster.

Cluster role

1. Créez un rôle de cluster si vous n'en avez pas déjà un pour accéder aux métriques :

```
$ cat <<EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus-scraper
rules:
- apiGroups:
  - image.openshift.io
  resources:
  - registry/metrics
  verbs:
  - get
EOF
```

2. Pour ajouter ce rôle à un utilisateur, exécutez la commande suivante :

```
$ oc adm policy add-cluster-role-to-user prometheus-scraper <username>
```

Metrics query

1. Obtenir le jeton de l'utilisateur.

```
openshift:
$ oc whoami -t
```

2. Exécuter une requête de métrique dans un nœud ou dans un pod, par exemple :

```
$ curl --insecure -s -u <user>:<secret> \ 1
    https://image-registry.openshift-image-registry.svc:5000/extensions/v2/metrics | grep
imageregistry | head -n 20
```

Exemple de sortie

```
# HELP imageregistry_build_info A metric with a constant '1' value labeled by major, minor,
git commit & git version from which the image registry was built.
# TYPE imageregistry_build_info gauge
imageregistry_build_info{gitCommit="9f72191",gitVersion="v3.11.0+9f72191-135-
dirty",major="3",minor="11+"} 1
# HELP imageregistry_digest_cache_requests_total Total number of requests without scope
to the digest cache.
# TYPE imageregistry_digest_cache_requests_total counter
imageregistry_digest_cache_requests_total{type="Hit"} 5
imageregistry_digest_cache_requests_total{type="Miss"} 24
# HELP imageregistry_digest_cache_scoped_requests_total Total number of scoped
requests to the digest cache.
# TYPE imageregistry_digest_cache_scoped_requests_total counter
imageregistry_digest_cache_scoped_requests_total{type="Hit"} 33
imageregistry_digest_cache_scoped_requests_total{type="Miss"} 44
# HELP imageregistry_http_in_flight_requests A gauge of requests currently being served by
the registry.
# TYPE imageregistry_http_in_flight_requests gauge
imageregistry_http_in_flight_requests 1
# HELP imageregistry_http_request_duration_seconds A histogram of latencies for requests
to the registry.
# TYPE imageregistry_http_request_duration_seconds summary
imageregistry_http_request_duration_seconds{method="get",quantile="0.5"} 0.01296087
imageregistry_http_request_duration_seconds{method="get",quantile="0.9"} 0.014847248
imageregistry_http_request_duration_seconds{method="get",quantile="0.99"} 0.015981195
imageregistry_http_request_duration_seconds_sum{method="get"} 12.260727916000022
```

1 L'objet **<user>** peut être arbitraire, mais la balise **<secret>** doit utiliser le jeton d'utilisateur.

4.6. RESSOURCES COMPLÉMENTAIRES

- Pour plus d'informations sur l'autorisation pour les pods d'un projet de référencer des images dans un autre projet, voir Autoriser les pods à référencer des images à travers les projets .
- Une adresse **kubeadmin** peut accéder au registre jusqu'à ce qu'elle soit supprimée. Voir Suppression de l'utilisateur kubeadmin pour plus d'informations.
- Pour plus d'informations sur la configuration d'un fournisseur d'identité, voir Comprendre la configuration d'un fournisseur d'identité.

CHAPITRE 5. EXPOSER LE REGISTRE

Par défaut, le registre d'OpenShift Container Platform est sécurisé lors de l'installation du cluster de sorte qu'il sert le trafic via TLS. Contrairement aux versions précédentes d'OpenShift Container Platform, le registre n'est pas exposé à l'extérieur du cluster au moment de l'installation.

5.1. EXPOSER MANUELLEMENT UN REGISTRE PAR DÉFAUT

Au lieu de se connecter au registre par défaut d'OpenShift Container Platform depuis l'intérieur du cluster, vous pouvez obtenir un accès externe à celui-ci en l'exposant avec une route. Cet accès externe vous permet de vous connecter au registre depuis l'extérieur du cluster en utilisant l'adresse de la route et d'étiqueter et de pousser des images vers un projet existant en utilisant l'hôte de la route.

Prérequis :

- Les conditions préalables suivantes sont automatiquement remplies :
  - Déployer l'opérateur de registre.
  - Déployer l'opérateur d'entrée.
- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

Vous pouvez exposer l'itinéraire en utilisant le paramètre **defaultRoute** dans la ressource **configs.imageregistry.operator.openshift.io**.

Pour exposer le registre à l'aide de **defaultRoute**:

1. Définir **defaultRoute** à **true**:

```
$ oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec":
{"defaultRoute":true}}' --type=merge
```

2. Obtenir l'itinéraire par défaut du registre :

```
$ HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
```

3. Obtenir le certificat de l'opérateur d'entrée :

```
$ oc get secret -n openshift-ingress  router-certs-default -o go-template='{{index .data
"tls.crt"}}' | base64 -d | sudo tee /etc/pki/ca-trust/source/anchors/${HOST}.crt  > /dev/null
```

4. Activez le certificat par défaut du cluster pour qu'il fasse confiance à l'itinéraire à l'aide des commandes suivantes :

```
$ sudo update-ca-trust enable
```

5. Connectez-vous avec podman en utilisant la route par défaut :

```
$ sudo podman login -u kubeadmin -p $(oc whoami -t) $HOST
```

5.2. EXPOSER MANUELLEMENT UN REGISTRE SÉCURISÉ

Au lieu de se connecter au registre d'OpenShift Container Platform depuis l'intérieur du cluster, vous pouvez obtenir un accès externe à celui-ci en l'exposant avec une route. Cela vous permet de vous connecter au registre depuis l'extérieur du cluster en utilisant l'adresse de la route, et d'étiqueter et de pousser des images vers un projet existant en utilisant l'hôte de la route.

Prérequis :

- Les conditions préalables suivantes sont automatiquement remplies :
  - Déployer l'opérateur de registre.
  - Déployer l'opérateur d'entrée.
- Vous avez accès au cluster en tant qu'utilisateur ayant le rôle **cluster-admin**.

Procédure

Vous pouvez exposer l'itinéraire en utilisant le paramètre **DefaultRoute** dans la ressource **configs.imageregistry.operator.openshift.io** ou en utilisant des itinéraires personnalisés.

Pour exposer le registre à l'aide de **DefaultRoute**:

1. Définir **DefaultRoute** à **True**:

```
$ oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec":
{"defaultRoute":true}}' --type=merge
```

2. Connectez-vous avec **podman**:

```
$ HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
```

```
$ podman login -u kubeadmin -p $(oc whoami -t) --tls-verify=false $HOST (1)
```

(1) **--tls-verify=false** est nécessaire si le certificat par défaut du cluster pour les itinéraires n'est pas fiable. Vous pouvez définir un certificat de confiance personnalisé comme certificat par défaut avec l'opérateur d'entrée.

Pour exposer le registre à l'aide de routes personnalisées :

1. Créez un secret avec les clés TLS de votre itinéraire :

```
$ oc create secret tls public-route-tls \
    -n openshift-image-registry \
    --cert=</path/to/tls.crt> \
    --key=</path/to/tls.key>
```

Cette étape est facultative. Si vous ne créez pas de secret, l'itinéraire utilise la configuration TLS par défaut de l'opérateur d'entrée.

2. Sur l'opérateur de registre :

```
spec:
```

```
  routes:
    - name: public-routes
      hostname: myregistry.mycorp.organization
      secretName: public-route-tls
...
```

NOTE

Ne définissez **secretName** que si vous fournissez une configuration TLS personnalisée pour la route du registre.