Ajouter un cluster

Astra Control Service

NetApp
June 04, 2024

This PDF was generated from https://docs.netapp.com/fr-fr/astra-control-service/get-started/install-astra-connector-previous.html on June 04, 2024. Always check docs.netapp.com for the latest.

Sommaire

Ajoutez un cluster à Astra Control Service . . . 1
Installez Astra Connector pour gérer les clusters . . . 1
Ajoutez un cluster géré par le fournisseur . . . 7
Ajoutez un cluster autogéré . . . 18

Ajoutez un cluster à Astra Control Service

Une fois votre environnement configuré, vous êtes prêt à créer un cluster Kubernetes, puis à l'ajouter à Astra Control Service. Vous pouvez ainsi utiliser Astra Control Service pour protéger vos applications sur le cluster.

Selon le type de cluster que vous devez ajouter à Astra Control Service, vous devez suivre différentes étapes pour ajouter le cluster.

- "Ajoutez un cluster géré par un fournisseur public à Astra Control Service": Utilisez ces étapes pour ajouter un cluster qui a une adresse IP publique et qui est géré par un fournisseur de cloud. Vous aurez besoin du compte principal de service, du compte de service ou du compte utilisateur du fournisseur cloud.
- "Ajoutez un cluster géré par un fournisseur privé à Astra Control Service": Procédez comme suit pour ajouter un cluster qui a une adresse IP privée et qui est géré par un fournisseur de cloud. Vous aurez besoin du compte principal de service, du compte de service ou du compte utilisateur du fournisseur cloud.
- "Ajoutez un cluster public autogéré à Astra Control Service": Utilisez ces étapes pour ajouter un cluster qui a une adresse IP publique et qui est géré par votre organisation. Vous devrez créer un fichier kubeconfig pour le cluster que vous souhaitez ajouter.
- "Ajoutez un cluster privé autogéré à Astra Control Service": Utilisez ces étapes pour ajouter un cluster qui a une adresse IP privée et qui est géré par votre organisation. Vous devrez créer un fichier kubeconfig pour le cluster que vous souhaitez ajouter.

Installez Astra Connector pour gérer les clusters

ASTRA Connector est un logiciel qui réside sur vos clusters gérés et qui facilite la communication entre le cluster géré et Astra Control. Pour les clusters gérés à l'aide d'Astra Control Service, deux versions d'Astra Connector sont disponibles :

- **Version précédente du connecteur Astra** : "Installer la version précédente du connecteur Astra" Sur votre cluster si vous prévoyez de gérer un cluster avec des workflows non natifs Kubernetes.
- [Aperçu technique] **Declarative Kubernetes Astra Connector** : "Installez Astra Connector pour les clusters gérés avec des workflows Kubernetes déclaratifs" Sur votre cluster si vous prévoyez de gérer le cluster à l'aide de workflows Kubernetes déclaratifs. Une fois que vous avez installé Astra Connector sur votre cluster, celui-ci est automatiquement ajouté à Astra Control.

> Le connecteur Kubernetes Astra déclaratif est disponible uniquement dans le cadre du programme Astra Control Early Adopter Program (EAP). Pour plus d'informations sur l'adhésion au programme EAP, contactez votre ingénieur commercial NetApp.

Installer la version précédente du connecteur Astra

ASTRA Control Service utilise la version précédente d'Astra Connector pour permettre la communication entre Astra Control Service et les clusters privés gérés avec des workflows non natifs Kubernetes. Vous devez installer Astra Connector sur des clusters privés que vous souhaitez gérer avec des workflows non natifs Kubernetes.

La version précédente d'Astra Connector prend en charge les types suivants de clusters privés gérés avec des workflows non natifs Kubernetes :

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift Service sur AWS (ROSA)
- ROSA avec AWS PrivateLink
- Red Hat OpenShift Container Platform sur site

Description de la tâche

- Lorsque vous effectuez ces étapes, exécutez ces commandes sur le cluster privé que vous souhaitez gérer avec Astra Control Service.
- Si vous utilisez un hôte bastion, exécutez ces commandes à partir de la ligne de commande de l'hôte bastion.

Avant de commencer

- Vous devez accéder au cluster privé que vous souhaitez gérer avec Astra Control Service.
- Vous devez disposer des autorisations d'administrateur Kubernetes pour installer l'opérateur Astra Connector sur le cluster.

Étapes

1. Installez l'ancien opérateur Astra Connector sur le cluster privé que vous souhaitez gérer avec des workflows non natifs Kubernetes. Lorsque vous exécutez cette commande, le namespace `astra-connector-operator` est créé et la configuration est appliquée au namespace :

   ```
   kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/23.07.0-202310251519/astraconnector_operator.yaml
   ```

2. Vérifiez que l'opérateur est installé et prêt :

   ```
   kubectl get all -n astra-connector-operator
   ```

3. Obtenez un jeton API d'Astra Control. Reportez-vous à la "Documentation relative à l'automatisation d'Astra" pour obtenir des instructions.
4. Créez le namespace astra-Connector :

   ```
   kubectl create ns astra-connector
   ```

5. Créez le fichier CR du connecteur Astra et nommez-le `astra-connector-cr.yaml`. Mettez à jour les valeurs entre parenthèses <> pour correspondre à votre environnement Astra Control et à la configuration du cluster :
   - **<ASTRA_CONTROL_SERVICE_URL>** : l'URL de l'interface utilisateur Web d'Astra Control Service. Par exemple :

```
https://astra.netapp.io
```

- **<ASTRA_CONTROL_SERVICE_API_TOKEN>** : jeton API Astra Control que vous avez obtenu à l'étape précédente.
- **<PRIVATE_AKS_CLUSTER_NAME>**: (Clusters AKS uniquement) - le nom du cluster du cluster privé Azure Kubernetes Service. Supprimez le commentaire et remplissez cette ligne uniquement si vous ajoutez un cluster AKS privé.
- **<ASTRA_CONTROL_ACCOUNT_ID>** : obtenu à partir de l'interface utilisateur Web d'Astra Control. Sélectionnez l'icône de figure en haut à droite de la page et sélectionnez **API Access**.

```
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. Après avoir renseigné le `astra-connector-cr.yaml` Fichier avec les valeurs correctes, appliquer la CR :

```
kubectl apply -f astra-connector-cr.yaml
```

7. Vérifier que le connecteur Astra est entièrement déployé :

```
kubectl get all -n astra-connector
```

8. Vérifier que le cluster est enregistré avec Astra Control :

```
kubectl get astraconnector -n astra-connector
```

Vous devez voir les résultats similaires à ce qui suit :

```
NAME              REGISTERED   ASTRACONNECTORID
STATUS
astra-connector   true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

Notez l'ASTRACONNECTORID ; vous en aurez besoin lorsque vous ajouterez le cluster à Astra Control.

Et la suite ?

Maintenant que vous avez installé Astra Connector, vous êtes prêt à ajouter votre cluster privé à Astra Control Service.

- "Ajoutez un cluster géré par un fournisseur privé à Astra Control Service": Procédez comme suit pour ajouter un cluster qui a une adresse IP privée et qui est géré par un fournisseur de cloud. Vous aurez besoin du compte principal de service, du compte de service ou du compte utilisateur du fournisseur cloud.
- "Ajoutez un cluster privé autogéré à Astra Control Service": Utilisez ces étapes pour ajouter un cluster qui a une adresse IP privée et qui est géré par votre organisation. Vous devrez créer un fichier kubeconfig pour le cluster que vous souhaitez ajouter.

Pour en savoir plus

- "Ajouter un cluster"

(Préversion technique) installez le connecteur Kubernetes Astra déclaratif

Les clusters gérés à l'aide de workflows Kubernetes déclaratifs utilisent Astra Connector pour permettre la communication entre le cluster géré et Astra Control. Vous devez installer Astra Connector sur tous les clusters que vous allez gérer avec des workflows Kubernetes déclaratifs.

Vous installez le connecteur Kubernetes Astra déclaratif à l'aide des commandes Kubernetes et des fichiers de ressources personnalisées (CR).

Description de la tâche

- Lorsque vous effectuez ces étapes, exécutez ces commandes sur le cluster que vous souhaitez gérer avec Astra Control.
- Si vous utilisez un hôte bastion, exécutez ces commandes à partir de la ligne de commande de l'hôte bastion.

Avant de commencer

- Vous devez accéder au cluster que vous souhaitez gérer avec Astra Control.
- Vous devez disposer des autorisations d'administrateur Kubernetes pour installer l'opérateur Astra Connector sur le cluster.

Si le cluster est configuré avec l'application d'admission de la sécurité du pod, c'est-à-dire la configuration par défaut pour les clusters Kubernetes 1.25 et versions ultérieures, vous devez activer les restrictions PSA sur les espaces de noms appropriés. Reportez-vous à la section "Préparez votre environnement à la gestion des clusters avec Astra Control" pour obtenir des instructions.

Étapes

1. Installez l'opérateur Astra Connector sur le cluster que vous souhaitez gérer avec des workflows Kubernetes déclaratifs. Lorsque vous exécutez cette commande, le namespace `astra-connector-operator` est créé et la configuration est appliquée au namespace :

```
kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/24.02.0-202403151353/astraconnector_operator.yaml
```

2. Vérifiez que l'opérateur est installé et prêt :

```
kubectl get all -n astra-connector-operator
```

3. Obtenez un jeton API d'Astra Control. Reportez-vous à la "Documentation relative à l'automatisation d'Astra" pour obtenir des instructions.
4. Créez un secret à l'aide du jeton. Remplacez <API_TOKEN> par le jeton que vous avez reçu d'Astra Control :

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Créez un secret Docker à utiliser pour extraire l'image du connecteur Astra. Remplacez les valeurs entre parenthèses <> par les informations de votre environnement :

<ASTRA_CONTROL_ACCOUNT_ID> est disponible dans l'interface utilisateur web d'Astra Control. Dans l'interface utilisateur Web, sélectionnez l'icône figure en haut à droite de la page et sélectionnez **accès API**.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Créez le fichier CR du connecteur Astra et nommez-le `astra-connector-cr.yaml`. Mettez à jour les valeurs entre parenthèses <> pour correspondre à votre environnement Astra Control et à la configuration du cluster :

- <ASTRA_CONTROL_ACCOUNT_ID> : obtenu à partir de l'interface utilisateur web d'Astra Control au cours de l'étape précédente.
- <CLUSTER_NAME> : nom que ce cluster doit être attribué dans Astra Control.
- <ASTRA_CONTROL_URL> : l'URL de l'interface utilisateur Web d'Astra Control. Par exemple :

```
https://astra.control.url
```

```
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. Après avoir renseigné le `astra-connector-cr.yaml` Fichier avec les valeurs correctes, appliquer la CR :

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Vérifier que le connecteur Astra est entièrement déployé :

```
kubectl get all -n astra-connector
```

9. Vérifier que le cluster est enregistré avec Astra Control :

```
kubectl get astraconnectors.astra.netapp.io -A
```

Vous devez voir les résultats similaires à ce qui suit :

```
NAMESPACE         NAME              REGISTERED   ASTRACONNECTORID
STATUS
astra-connector   astra-connector   true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Vérifiez que le cluster s'affiche dans la liste des clusters gérés sur la page **clusters** de l'interface utilisateur Web d'Astra Control.

Ajoutez un cluster géré par le fournisseur

Ajoutez un cluster géré par un fournisseur public à Astra Control Service

Une fois que vous avez configuré votre environnement cloud, vous êtes prêt à créer un cluster Kubernetes, puis à l'ajouter à Astra Control Service.

- Créez un cluster Kubernetes
- Ajoutez le cluster à Astra Control Service
- Modifiez la classe de stockage par défaut

Créez un cluster Kubernetes

Si vous ne disposez pas encore d'un cluster, vous pouvez en créer un qui rencontre "Exigences d'Astra Control Service pour Amazon Elastic Kubernetes Service (EKS)". Si vous ne disposez pas encore d'un cluster, vous pouvez en créer un qui rencontre "Exigences d'Astra Control Service pour Google Kubernetes Engine (GKE)". Si vous ne disposez pas encore d'un cluster, vous pouvez en créer un qui rencontre "Astra Control Service exigences pour Azure Kubernetes Service (AKS) avec Azure NetApp Files" ou "Astra Control Service exigences pour Azure Kubernetes Service (AKS) avec des disques gérés Azure".

Astra Control Service prend en charge les clusters AKS qui utilisent Azure Active Directory (Azure AD) pour l'authentification et la gestion des identités. Une fois le cluster créé, suivez les instructions du "documentation officielle" Pour configurer le cluster afin d'utiliser Azure AD. Vous devez vous assurer que vos clusters répondent aux exigences de l'intégration d'Azure AD gérée par AKS.

Ajoutez le cluster à Astra Control Service

Une fois connecté au service Astra Control, la première étape consiste à commencer à gérer vos clusters. Avant d'ajouter un cluster à Astra Control Service, vous devez effectuer des tâches spécifiques et vous assurer qu'il répond à certaines exigences.

Lorsque vous gérez des clusters Azure Kubernetes Service et Google Kubernetes Engine, notez que vous disposez de deux options pour l'installation et la gestion du cycle de vie d'Astra Control Provisioner :

- Vous pouvez utiliser Astra Control Service pour gérer automatiquement le cycle de vie d'Astra Control Provisioner. Pour ce faire, assurez-vous qu'Astra Trident n'est pas installée et que Astra Control Provisioner n'est pas activé sur le cluster que vous souhaitez gérer avec Astra Control Service. Dans ce cas, Astra Control Service active automatiquement Astra Control provisionner lorsque vous commencez à gérer le cluster, et les mises à niveau d'Astra Control provisionner sont gérées automatiquement.

- Vous pouvez gérer vous-même le cycle de vie d'Astra Control Provisioner. Pour ce faire, activez Astra Control Provisioner sur le cluster avant de gérer le cluster avec Astra Control Service. Dans ce cas, Astra Control Service détecte que le mécanisme de provisionnement Astra Control est déjà activé et ne le réinstalle pas et ne gère pas les mises à niveau d'Astra Control Provisioner. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" Pour connaître les étapes à suivre, activez le mécanisme de provisionnement Astra Control.

Lorsque vous gérez des clusters Amazon Web Services avec Astra Control Service, si vous avez besoin de systèmes back-end de stockage qui ne peuvent être utilisés qu'avec Astra Control provisionner, vous devez activer manuellement le mécanisme Astra Control provisionner sur le cluster avant de le gérer avec Astra Control Service. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" Pour connaître les étapes à suivre pour activer le mécanisme de provisionnement Astra Control.

Avant de commencer

Amazon Web Services

- Vous devez disposer du fichier JSON contenant les informations d'identification de l'utilisateur IAM qui a créé le cluster. "Découvrez comment créer un utilisateur IAM".
- ASTRA Control Provisioner est requis pour Amazon FSX pour NetApp ONTAP. Si vous prévoyez d'utiliser Amazon FSX pour NetApp ONTAP en tant que back-end de stockage pour votre cluster EKS, reportez-vous aux informations concernant Astra Control provisionner du "Configuration requise pour le cluster EKS".
- (Facultatif) si vous devez fournir les informations nécessaires `kubectl` L'accès aux commandes d'un cluster à d'autres utilisateurs IAM qui ne sont pas le créateur du cluster, reportez-vous aux instructions de la "Comment puis-je fournir l'accès aux autres utilisateurs IAM et aux rôles après la création du cluster dans Amazon EKS ?".
- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Amazon Web Services. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Microsoft Azure

- Vous devez disposer du fichier JSON qui contient la sortie de l'interface de ligne de commandes Azure lorsque vous avez créé le principal de service. "Découvrez comment configurer un principal de service".

  Vous aurez également besoin de votre ID d'abonnement Azure, si vous n'avez pas ajouté le fichier JSON.

- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Microsoft Azure. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Google Cloud

- Vous devez disposer du fichier de clé de compte de service pour un compte de service disposant des autorisations requises. "Découvrez comment configurer un compte de service".
- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Google Cloud. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Étapes

1. (Facultatif) si vous ajoutez un cluster Amazon EKS ou si vous souhaitez gérer vous-même l'installation et les mises à niveau d'Astra Control Provisioner, activez Astra Control Provisioner sur le cluster. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" des étapes d'accompagnement.
2. Ouvrez l'interface utilisateur web d'Astra Control Service dans un navigateur.
3. Dans le Tableau de bord, sélectionnez **Manage Kubernetes cluster**.

   Suivez les invites pour ajouter le cluster.

4. **Fournisseur** : sélectionnez votre fournisseur de cloud, puis fournissez les informations d'identification requises pour créer une nouvelle instance de cloud ou sélectionnez une instance de cloud existante à utiliser.
5. **Amazon Web Services**: Fournissez des détails sur votre compte utilisateur Amazon Web Services IAM en téléchargeant un fichier JSON ou en collant le contenu de ce fichier JSON à partir de votre presse-papiers.

   Le fichier JSON doit contenir les informations d'identification de l'utilisateur IAM qui a créé le cluster.

6. **Microsoft Azure**: Fournissez des détails sur votre entité de service Azure en téléchargeant un fichier JSON ou en collant le contenu de ce fichier JSON à partir de votre presse-papiers.

   Le fichier JSON doit contenir la sortie de l'interface de ligne de commandes Azure lorsque vous avez créé le principal de service. Il peut également inclure votre identifiant d'abonnement afin qu'il soit automatiquement ajouté à Astra. Sinon, vous devez saisir manuellement l'ID après avoir fourni le fichier JSON.

7. **Google Cloud Platform**: Fournir le fichier de clé de compte de service soit en téléchargeant le fichier ou en collant le contenu à partir de votre presse-papiers.

   Astra Control Service utilise le compte de service pour détecter les clusters qui s'exécutent dans Google Kubernetes Engine.

8. **Autre** : cet onglet est destiné uniquement aux clusters autogérés.
   a. **Nom de l'instance de Cloud** : indiquez un nom pour la nouvelle instance de Cloud qui sera créée lorsque vous ajoutez ce cluster. En savoir plus sur "instances cloud".
   b. Sélectionnez **Suivant**.

      ASTRA Control Service affiche la liste des clusters disponibles.

   c. **Cluster** : sélectionnez un cluster dans la liste à ajouter à Astra Control Service.

      > Lorsque vous sélectionnez dans la liste des groupes, faites attention à la colonne **Eligiblity**. Si un cluster est « inéligible » ou « partiellement éligible », passez la souris sur l'état pour déterminer s'il y a un problème au niveau du cluster. Par exemple, il peut identifier que le cluster ne dispose pas d'un nœud worker.

   d. Sélectionnez **Suivant**.
   e. (Facultatif) **Storage** : si vous le souhaitez, sélectionnez la classe de stockage que les applications Kubernetes déployées sur ce cluster doivent utiliser par défaut.
9. Pour sélectionner une nouvelle classe de stockage par défaut pour le cluster, cochez la case **affecter une nouvelle classe de stockage par défaut**.
10. Sélectionnez une nouvelle classe de stockage par défaut dans la liste.

Chaque fournisseur de service de stockage cloud affiche les informations suivantes en matière de prix, de performance et de résilience :

- Cloud Volumes Service pour Google Cloud : informations sur le prix, la performance et la résilience
- Google persistent Disk : pas d'informations sur le prix, la performance ou la résilience disponibles
- Azure NetApp Files : informations sur les performances et la résilience
- Azure Managed Disks : aucun prix, performances ou résilience disponibles
- Amazon Elastic Block Store : pas d'informations disponibles sur le prix, la performance ou la résilience
- Amazon FSX pour NetApp ONTAP : aucune information disponible concernant le prix, les performances ou la résilience
- NetApp Cloud Volumes ONTAP : aucune information disponible sur le prix, les performances ou la résilience

Chaque classe de stockage peut utiliser l'un des services suivants :

- "Cloud Volumes Service pour Google Cloud"
- "Disque persistant Google"
  - "Azure NetApp Files"
  - "Disques gérés Azure"
  - "Amazon Elastic Block Store"
  - "Amazon FSX pour NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

    En savoir plus sur "Classes de stockage pour les clusters Amazon Web Services". En savoir plus sur "Classes de stockage pour les clusters AKS". En savoir plus sur "Classes de stockage pour clusters GKE".

    a. Sélectionnez **Suivant**.
    b. **Revoir et approuver** : consultez les détails de la configuration.
    c. Sélectionnez **Ajouter** pour ajouter le cluster à Astra Control Service.

Résultat

S'il s'agit du premier cluster que vous avez ajouté pour ce fournisseur cloud, Astra Control Service crée un magasin d'objets pour le fournisseur cloud pour les sauvegardes d'applications s'exécutant sur les clusters éligibles. (Lorsque vous ajoutez des clusters suivants pour ce fournisseur de cloud, aucun magasin d'objets n'est créé.) Si vous avez spécifié une classe de stockage par défaut, Astra Control Service définit la classe de stockage par défaut que vous avez spécifiée. Pour les clusters gérés dans Amazon Web Services ou Google Cloud Platform, Astra Control Service crée également un compte d'administration sur le cluster. Ces actions peuvent prendre plusieurs minutes.

Modifiez la classe de stockage par défaut

Vous pouvez modifier la classe de stockage par défaut d'un cluster.

Modifiez la classe de stockage par défaut avec Astra Control

Vous pouvez modifier la classe de stockage par défaut d'un cluster depuis Astra Control. Si votre cluster utilise un service back-end de stockage installé précédemment, il se peut que vous ne puissiez pas utiliser cette méthode pour modifier la classe de stockage par défaut (l'action **Set as default** n'est pas sélectionnable). Dans ce cas, vous pouvez Modifiez la classe de stockage par défaut à l'aide de la ligne de commande.

Étapes

1. Dans l'interface utilisateur du service de contrôle Astra, sélectionnez **clusters**.
2. Sur la page **clusters**, sélectionnez le cluster que vous souhaitez modifier.
3. Sélectionnez l'onglet **stockage**.
4. Sélectionnez la catégorie **classes de stockage**.
5. Sélectionnez le menu **actions** pour la classe de stockage que vous souhaitez définir par défaut.
6. Sélectionnez **définir comme valeur par défaut**.

Modifiez la classe de stockage par défaut à l'aide de la ligne de commande

Vous pouvez modifier la classe de stockage par défaut d'un cluster à l'aide des commandes Kubernetes. Cette méthode fonctionne quelle que soit la configuration du cluster.

Étapes

1. Connectez-vous à votre cluster Kubernetes.
2. Lister les classes de stockage de votre cluster :

```
kubectl get storageclass
```

3. Supprimez la désignation par défaut de la classe de stockage par défaut. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Sélectionnez par défaut une classe de stockage différente. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirmez la nouvelle classe de stockage par défaut :

```
kubectl get storageclass
```

Ajoutez un cluster géré par un fournisseur privé à Astra Control Service

Vous pouvez utiliser Astra Control Service pour gérer les clusters Google Kubernetes Engine (GKE) privés. Ces instructions supposent que vous avez déjà créé un cluster AKS ou OpenShift privé et préparé une méthode sécurisée pour y accéder à distance. Pour plus d'informations sur la création et l'accès à des clusters AKS ou OpenShift privés, reportez-vous à la documentation suivante :

- "Documentation Azure pour clusters AKS privés"
- "Documentation Azure pour les clusters OpenShift privés"

Vous pouvez utiliser Astra Control Service pour gérer des clusters Azure Kubernetes Service (AKS) privés ainsi que des clusters Red Hat OpenShift privés dans AKS. Ces instructions supposent que vous avez déjà créé un cluster AKS ou OpenShift privé et préparé une méthode sécurisée pour y accéder à distance. Pour plus d'informations sur la création et l'accès à des clusters AKS ou OpenShift privés, reportez-vous à la documentation suivante :

- "Documentation Azure pour clusters AKS privés"
- "Documentation Azure pour les clusters OpenShift privés"

Vous pouvez utiliser Astra Control Service pour gérer les clusters Amazon Elastic Kubernetes Service (EKS) privés. Ces instructions supposent que vous avez déjà créé un cluster EKS privé et préparé une méthode sécurisée pour y accéder à distance. Pour plus d'informations sur la création et l'accès à des clusters EKS privés, reportez-vous au "Documentation Amazon EKS".

Pour ajouter votre cluster privé à Astra Control Service, vous devez effectuer les tâches suivantes :

1. Poser le connecteur Astra
2. Configuration du stockage persistant
3. Ajoutez le cluster géré par le fournisseur privé à Astra Control Service

Poser le connecteur Astra

Avant d'ajouter un cluster privé, vous devez installer Astra Connector sur le cluster afin qu'Astra Control puisse communiquer avec lui. Reportez-vous à la section "Installez la version précédente d'Astra Connector pour les clusters privés gérés avec des workflows non natifs Kubernetes" pour obtenir des instructions.

Configuration du stockage persistant

Configurer le stockage persistant pour le cluster. Pour plus d'informations sur la configuration du stockage persistant, reportez-vous à la documentation de mise en route :

- "Configuration de Microsoft Azure avec Azure NetApp Files"
- "Configuration de Microsoft Azure avec des disques gérés Azure"
- "Configurer Amazon Web Services"
- "Configurez Google Cloud"

Ajoutez le cluster géré par le fournisseur privé à Astra Control Service

Vous pouvez maintenant ajouter le cluster privé à Astra Control Service.

Lorsque vous gérez des clusters Azure Kubernetes Service et Google Kubernetes Engine, notez que vous disposez de deux options pour l'installation et la gestion du cycle de vie d'Astra Control Provisioner :

- Vous pouvez utiliser Astra Control Service pour gérer automatiquement le cycle de vie d'Astra Control Provisioner. Pour ce faire, assurez-vous qu'Astra Trident n'est pas installée et que Astra Control Provisioner n'est pas activé sur le cluster que vous souhaitez gérer avec Astra Control Service. Dans ce cas, Astra Control Service active automatiquement Astra Control provisionner lorsque vous commencez à gérer le cluster, et les mises à niveau d'Astra Control provisionner sont gérées automatiquement.
- Vous pouvez gérer vous-même le cycle de vie d'Astra Control Provisioner. Pour ce faire, activez Astra Control Provisioner sur le cluster avant de gérer le cluster avec Astra Control Service. Dans ce cas, Astra Control Service détecte que le mécanisme de provisionnement Astra Control est déjà activé et ne le réinstalle pas et ne gère pas les mises à niveau d'Astra Control Provisioner. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" Pour connaître les étapes à suivre, activez le mécanisme de provisionnement Astra Control.

Lorsque vous gérez des clusters Amazon Web Services avec Astra Control Service, si vous avez besoin de systèmes back-end de stockage qui ne peuvent être utilisés qu'avec Astra Control provisionner, vous devez activer manuellement le mécanisme Astra Control provisionner sur le cluster avant de le gérer avec Astra Control Service. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" Pour connaître les étapes à suivre pour activer le mécanisme de provisionnement Astra Control.

Avant de commencer

Amazon Web Services

- Vous devez disposer du fichier JSON contenant les informations d'identification de l'utilisateur IAM qui a créé le cluster. "Découvrez comment créer un utilisateur IAM".
- ASTRA Control Provisioner est requis pour Amazon FSX pour NetApp ONTAP. Si vous prévoyez d'utiliser Amazon FSX pour NetApp ONTAP en tant que back-end de stockage pour votre cluster EKS, reportez-vous aux informations concernant Astra Control provisionner du "Configuration requise pour le cluster EKS".
- (Facultatif) si vous devez fournir les informations nécessaires `kubectl` L'accès aux commandes d'un cluster à d'autres utilisateurs IAM qui ne sont pas le créateur du cluster, reportez-vous aux instructions de la "Comment puis-je fournir l'accès aux autres utilisateurs IAM et aux rôles après la création du cluster dans Amazon EKS ?".
- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Amazon Web Services. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Microsoft Azure

- Vous devez disposer du fichier JSON qui contient la sortie de l'interface de ligne de commandes Azure lorsque vous avez créé le principal de service. "Découvrez comment configurer un principal de service".

  Vous aurez également besoin de votre ID d'abonnement Azure, si vous n'avez pas ajouté le fichier JSON.

- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Microsoft Azure. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Google Cloud

- Vous devez disposer du fichier de clé de compte de service pour un compte de service disposant des autorisations requises. "Découvrez comment configurer un compte de service".
- Si le cluster est privé, le "réseaux autorisés" Doit autoriser l'adresse IP du service de contrôle Astra :

  52.188.218.166/32

- Si vous prévoyez d'utiliser NetApp Cloud Volumes ONTAP en tant que système back-end de stockage, vous devez configurer Cloud Volumes ONTAP pour qu'il fonctionne avec Google Cloud. Consultez le Cloud Volumes ONTAP "documentation de configuration".

Étapes

1. (Facultatif) si vous ajoutez un cluster Amazon EKS ou si vous souhaitez gérer vous-même l'installation et les mises à niveau d'Astra Control Provisioner, activez Astra Control Provisioner sur le cluster. Reportez-vous à la section "Activez le mécanisme de provisionnement Astra Control" des étapes d'accompagnement.
2. Ouvrez l'interface utilisateur web d'Astra Control Service dans un navigateur.
3. Dans le Tableau de bord, sélectionnez **Manage Kubernetes cluster**.

   Suivez les invites pour ajouter le cluster.

4. **Fournisseur** : sélectionnez votre fournisseur de cloud, puis fournissez les informations d'identification requises pour créer une nouvelle instance de cloud ou sélectionnez une instance de cloud existante à utiliser.
5. **Amazon Web Services**: Fournissez des détails sur votre compte utilisateur Amazon Web Services IAM en téléchargeant un fichier JSON ou en collant le contenu de ce fichier JSON à partir de votre presse-papiers.

   Le fichier JSON doit contenir les informations d'identification de l'utilisateur IAM qui a créé le cluster.

6. **Microsoft Azure**: Fournissez des détails sur votre entité de service Azure en téléchargeant un fichier JSON ou en collant le contenu de ce fichier JSON à partir de votre presse-papiers.

   Le fichier JSON doit contenir la sortie de l'interface de ligne de commandes Azure lorsque vous avez créé le principal de service. Il peut également inclure votre identifiant d'abonnement afin qu'il soit automatiquement ajouté à Astra. Sinon, vous devez saisir manuellement l'ID après avoir fourni le fichier JSON.

7. **Google Cloud Platform**: Fournir le fichier de clé de compte de service soit en téléchargeant le fichier ou en collant le contenu à partir de votre presse-papiers.

   Astra Control Service utilise le compte de service pour détecter les clusters qui s'exécutent dans Google Kubernetes Engine.

8. **Autre** : cet onglet est destiné uniquement aux clusters autogérés.
   a. **Nom de l'instance de Cloud** : indiquez un nom pour la nouvelle instance de Cloud qui sera créée lorsque vous ajoutez ce cluster. En savoir plus sur "instances cloud".
   b. Sélectionnez **Suivant**.

      ASTRA Control Service affiche la liste des clusters disponibles.

   c. **Cluster** : sélectionnez un cluster dans la liste à ajouter à Astra Control Service.

      Lorsque vous sélectionnez dans la liste des groupes, faites attention à la colonne **Eligiblity**. Si un cluster est « inéligible » ou « partiellement éligible », passez la souris sur l'état pour déterminer s'il y a un problème au niveau du cluster. Par exemple, il peut identifier que le cluster ne dispose pas d'un nœud worker.

9. Sélectionnez **Suivant**.
10. (Facultatif) **Storage** : si vous le souhaitez, sélectionnez la classe de stockage que les applications Kubernetes déployées sur ce cluster doivent utiliser par défaut.
    a. Pour sélectionner une nouvelle classe de stockage par défaut pour le cluster, cochez la case **affecter une nouvelle classe de stockage par défaut**.
    b. Sélectionnez une nouvelle classe de stockage par défaut dans la liste.

Chaque fournisseur de service de stockage cloud affiche les informations suivantes en matière de prix, de performance et de résilience :

- Cloud Volumes Service pour Google Cloud : informations sur le prix, la performance et la résilience
- Google persistent Disk : pas d'informations sur le prix, la performance ou la résilience disponibles
- Azure NetApp Files : informations sur les performances et la résilience
- Azure Managed Disks : aucun prix, performances ou résilience disponibles
- Amazon Elastic Block Store : pas d'informations disponibles sur le prix, la performance ou la résilience
- Amazon FSX pour NetApp ONTAP : aucune information disponible concernant le prix, les performances ou la résilience
- NetApp Cloud Volumes ONTAP : aucune information disponible sur le prix, les performances ou la résilience

Chaque classe de stockage peut utiliser l'un des services suivants :

- "Cloud Volumes Service pour Google Cloud"
- "Disque persistant Google"
- "Azure NetApp Files"
- "Disques gérés Azure"
- "Amazon Elastic Block Store"
- "Amazon FSX pour NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

En savoir plus sur "Classes de stockage pour les clusters Amazon Web Services". En savoir plus sur "Classes de stockage pour les clusters AKS". En savoir plus sur "Classes de stockage pour clusters GKE".

c. Sélectionnez **Suivant**.

d. **Revoir et approuver** : consultez les détails de la configuration.

e. Sélectionnez **Ajouter** pour ajouter le cluster à Astra Control Service.

Résultat

S'il s'agit du premier cluster que vous avez ajouté pour ce fournisseur cloud, Astra Control Service crée un magasin d'objets pour le fournisseur cloud pour les sauvegardes d'applications s'exécutant sur les clusters éligibles. (Lorsque vous ajoutez des clusters suivants pour ce fournisseur de cloud, aucun magasin d'objets n'est créé.) Si vous avez spécifié une classe de stockage par défaut, Astra Control Service définit la classe de stockage par défaut que vous avez spécifiée. Pour les clusters gérés dans Amazon Web Services ou Google Cloud Platform, Astra Control Service crée également un compte d'administration sur le cluster. Ces actions peuvent prendre plusieurs minutes.

Modifiez la classe de stockage par défaut

Vous pouvez modifier la classe de stockage par défaut d'un cluster.

Modifiez la classe de stockage par défaut avec Astra Control

Vous pouvez modifier la classe de stockage par défaut d'un cluster depuis Astra Control. Si votre cluster utilise un service back-end de stockage installé précédemment, il se peut que vous ne puissiez pas utiliser cette méthode pour modifier la classe de stockage par défaut (l'action **Set as default** n'est pas sélectionnable). Dans ce cas, vous pouvez Modifiez la classe de stockage par défaut à l'aide de la ligne de commande.

Étapes

1. Dans l'interface utilisateur du service de contrôle Astra, sélectionnez **clusters**.
2. Sur la page **clusters**, sélectionnez le cluster que vous souhaitez modifier.
3. Sélectionnez l'onglet **stockage**.
4. Sélectionnez la catégorie **classes de stockage**.
5. Sélectionnez le menu **actions** pour la classe de stockage que vous souhaitez définir par défaut.
6. Sélectionnez **définir comme valeur par défaut**.

Modifiez la classe de stockage par défaut à l'aide de la ligne de commande

Vous pouvez modifier la classe de stockage par défaut d'un cluster à l'aide des commandes Kubernetes. Cette méthode fonctionne quelle que soit la configuration du cluster.

Étapes

1. Connectez-vous à votre cluster Kubernetes.
2. Lister les classes de stockage de votre cluster :

```
kubectl get storageclass
```

3. Supprimez la désignation par défaut de la classe de stockage par défaut. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Sélectionnez par défaut une classe de stockage différente. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirmez la nouvelle classe de stockage par défaut :

```
kubectl get storageclass
```

Ajoutez un cluster autogéré

Ajoutez un cluster public autogéré à Astra Control Service

Une fois votre environnement configuré, vous êtes prêt à créer un cluster Kubernetes, puis à l'ajouter à Astra Control Service.

Un cluster autogéré est un cluster que vous provisionnez et gérez directement. ASTRA Control Service prend en charge les clusters autogérés qui s'exécutent dans un environnement de cloud public. Vous pouvez ajouter un cluster auto-géré au service Astra Control en téléchargeant un `kubeconfig.yaml` fichier. Vous devez vous assurer que le cluster répond aux exigences décrites ici.

Distributions Kubernetes prises en charge

Vous pouvez utiliser Astra Control Service pour gérer les types suivants de clusters publics et autogérés :

Distribution Kubernetes	Versions prises en charge
Kubernetes (en amont)	1.27 à 1.29
Rancher Kubernetes Engine (RKE)	RKE 1 : versions 1.24.17, 1.25.13, 1.26.8 avec Rancher Manager 2.7.9 RKE 2 : versions 1.23.16 et 1.24.13 avec Rancher Manager 2.6.13 RKE 2 : versions 1.24.17, 1.25.14, 1.26.9 avec Rancher Manager 2.7.9
Plateforme de conteneurs Red Hat OpenShift	4.12 à 4.14

Ces instructions supposent que vous avez déjà créé un cluster autogéré.

- Ajoutez le cluster à Astra Control Service
- Modifiez la classe de stockage par défaut

Ajoutez le cluster à Astra Control Service

Une fois connecté au service Astra Control, la première étape consiste à commencer à gérer vos clusters. Avant d'ajouter un cluster à Astra Control Service, vous devez effectuer des tâches spécifiques et vous assurer qu'il répond à certaines exigences.

Avant de commencer

Un cluster autogéré est un cluster que vous provisionnez et gérez directement. ASTRA Control Service prend en charge les clusters autogérés qui s'exécutent dans un environnement de cloud public. Vos clusters autogérés peuvent utiliser Astra Control Provisioner pour s'interfacer avec les services de stockage NetApp ou des pilotes Container Storage interface (CSI) pour s'interfacer avec Amazon Elastic Block Store (EBS), les disques gérés Azure et le service Google persistent Disk.

ASTRA Control Service prend en charge les clusters autogérés qui utilisent les distributions Kubernetes suivantes :

- Plateforme de conteneurs Red Hat OpenShift
- Moteur rancher Kubernetes
- Kubernetes en amont

Votre cluster autogéré doit répondre aux exigences suivantes :

- Le cluster doit être accessible via Internet.
- Si vous utilisez ou prévoyez d'utiliser le stockage activé avec des pilotes CSI, les pilotes CSI appropriés doivent être installés sur le cluster. Pour plus d'informations sur l'utilisation des pilotes CSI pour intégrer le stockage, reportez-vous à la documentation de votre service de stockage.
- Vous avez accès au fichier kubeconfig du cluster qui ne contient qu'un seul élément de contexte. Suivre "ces instructions" pour générer un fichier kubeconfig.
- Si vous ajoutez le cluster à l'aide d'un fichier kubeconfig qui fait référence à une autorité de certification privée (CA), ajoutez la ligne suivante au `cluster` section du fichier kubeconfig. Cela permet à Astra Control d'ajouter le cluster :

  ```
  insecure-skip-tls-verify: true
  ```

- **Rancher uniquement**: Lorsque vous gérez des clusters d'applications dans un environnement Rancher, modifiez le contexte par défaut du cluster d'applications dans le fichier kubeconfig fourni par Rancher pour utiliser un contexte de plan de contrôle au lieu du contexte du serveur d'API Rancher. La charge est réduite sur le serveur API Rancher et les performances sont améliorées.
- **Exigences du mécanisme de provisionnement Astra Control** : vous devez avoir un mécanisme de provisionnement Astra Control correctement configuré, y compris ses composants Astra Trident, pour gérer les clusters.
  - **Revoir les exigences de l'environnement Astra Trident** : avant d'installer ou de mettre à niveau Astra Control Provisioner, consultez le "systèmes front-end, systèmes back-end et configurations hôte pris en charge".
  - **Activer la fonctionnalité Astra Control Provisioner** : il est fortement recommandé d'installer Astra Trident 23.10 ou version ultérieure et de l'activer "Fonctionnalité de stockage avancée Astra Control Provisioner". Dans les prochaines versions, Astra Control ne prendra pas en charge Astra Trident si le mécanisme de provisionnement Astra Control n'est pas également activé.
  - **Configurer un back-end de stockage** : au moins un back-end de stockage doit l'être "Configuré dans Astra Trident" sur le cluster.
  - **Configurer une classe de stockage** : au moins une classe de stockage doit être "Configuré dans Astra Trident" sur le cluster. Si une classe de stockage par défaut est configurée, assurez-vous qu'il s'agit de la classe de stockage **Only** qui possède l'annotation par défaut.

- **Configurer un contrôleur de snapshot de volume et installer une classe de snapshot de volume** : "Installez un contrôleur de snapshot de volume" Il est ainsi possible de créer des snapshots dans Astra Control. "Création" au moins un `VolumeSnapshotClass` Avec Astra Trident.

Étapes

1. Dans le Tableau de bord, sélectionnez **Manage Kubernetes cluster**.

   Suivez les invites pour ajouter le cluster.

2. **Fournisseur** : sélectionnez l'onglet **autre** pour ajouter des détails sur votre cluster auto-géré.
   a. **Autre**: Fournir des détails sur votre cluster auto-géré en téléchargeant un `kubeconfig.yaml` ou en collant le contenu du `kubeconfig.yaml` fichier à partir du presse-papiers.

   > Si vous créez la vôtre `kubeconfig` fichier, vous ne devez définir que **un** élément de contexte dans celui-ci. Reportez-vous à la section "Documentation Kubernetes" pour plus d'informations sur la création `kubeconfig` fichiers.

3. **Nom d'identification** : indiquez un nom pour les informations d'identification de cluster autogérées que vous téléchargez sur Astra Control. Par défaut, le nom des identifiants est automatiquement renseigné comme nom du cluster.
4. **ID de route privée** : ce champ est destiné uniquement aux clusters privés.
5. Sélectionnez **Suivant**.
6. (Facultatif) **Storage** : si vous le souhaitez, sélectionnez la classe de stockage que les applications Kubernetes déployées sur ce cluster doivent utiliser par défaut.
   a. Pour sélectionner une nouvelle classe de stockage par défaut pour le cluster, cochez la case **affecter une nouvelle classe de stockage par défaut**.
   b. Sélectionnez une nouvelle classe de stockage par défaut dans la liste.

   > Chaque fournisseur de service de stockage cloud affiche les informations suivantes en matière de prix, de performance et de résilience :
   >
   > - Cloud Volumes Service pour Google Cloud : informations sur le prix, la performance et la résilience
   > - Google persistent Disk : pas d'informations sur le prix, la performance ou la résilience disponibles
   > - Azure NetApp Files : informations sur les performances et la résilience
   > - Azure Managed Disks : aucun prix, performances ou résilience disponibles
   > - Amazon Elastic Block Store : pas d'informations disponibles sur le prix, la performance ou la résilience
   > - Amazon FSX pour NetApp ONTAP : aucune information disponible concernant le prix, les performances ou la résilience
   > - NetApp Cloud Volumes ONTAP : aucune information disponible sur le prix, les performances ou la résilience

   Chaque classe de stockage peut utiliser l'un des services suivants :

- "Cloud Volumes Service pour Google Cloud"
- "Disque persistant Google"
  - "Azure NetApp Files"
  - "Disques gérés Azure"
  - "Amazon Elastic Block Store"
  - "Amazon FSX pour NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

    En savoir plus sur "Classes de stockage pour les clusters Amazon Web Services". En savoir plus sur "Classes de stockage pour les clusters AKS". En savoir plus sur "Classes de stockage pour clusters GKE".

c. Sélectionnez **Suivant**.

d. **Revoir et approuver** : consultez les détails de la configuration.

e. Sélectionnez **Ajouter** pour ajouter le cluster à Astra Control Service.

Modifiez la classe de stockage par défaut

Vous pouvez modifier la classe de stockage par défaut d'un cluster.

Modifiez la classe de stockage par défaut avec Astra Control

Vous pouvez modifier la classe de stockage par défaut d'un cluster depuis Astra Control. Si votre cluster utilise un service back-end de stockage installé précédemment, il se peut que vous ne puissiez pas utiliser cette méthode pour modifier la classe de stockage par défaut (l'action **Set as default** n'est pas sélectionnable). Dans ce cas, vous pouvez Modifiez la classe de stockage par défaut à l'aide de la ligne de commande.

Étapes

1. Dans l'interface utilisateur du service de contrôle Astra, sélectionnez **clusters**.
2. Sur la page **clusters**, sélectionnez le cluster que vous souhaitez modifier.
3. Sélectionnez l'onglet **stockage**.
4. Sélectionnez la catégorie **classes de stockage**.
5. Sélectionnez le menu **actions** pour la classe de stockage que vous souhaitez définir par défaut.
6. Sélectionnez **définir comme valeur par défaut**.

Modifiez la classe de stockage par défaut à l'aide de la ligne de commande

Vous pouvez modifier la classe de stockage par défaut d'un cluster à l'aide des commandes Kubernetes. Cette méthode fonctionne quelle que soit la configuration du cluster.

Étapes

1. Connectez-vous à votre cluster Kubernetes.
2. Lister les classes de stockage de votre cluster :

```
kubectl get storageclass
```

3. Supprimez la désignation par défaut de la classe de stockage par défaut. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Sélectionnez par défaut une classe de stockage différente. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirmez la nouvelle classe de stockage par défaut :

```
kubectl get storageclass
```

Ajoutez un cluster privé autogéré à Astra Control Service

Une fois votre environnement configuré, vous êtes prêt à créer un cluster Kubernetes, puis à l'ajouter à Astra Control Service.

Un cluster autogéré est un cluster que vous provisionnez et gérez directement. ASTRA Control Service prend en charge les clusters autogérés qui s'exécutent dans un environnement de cloud public. Vous pouvez ajouter un cluster auto-géré au service Astra Control en téléchargeant un `kubeconfig.yaml` fichier. Vous devez vous assurer que le cluster répond aux exigences décrites ici.

Distributions Kubernetes prises en charge

Vous pouvez utiliser Astra Control Service pour gérer les types suivants de clusters privés et autogérés :

Distribution Kubernetes	Versions prises en charge
Kubernetes (en amont)	1.27 à 1.29
Rancher Kubernetes Engine (RKE)	RKE 1 : versions 1.24.17, 1.25.13, 1.26.8 avec Rancher Manager 2.7.9 RKE 2 : versions 1.23.16 et 1.24.13 avec Rancher Manager 2.6.13 RKE 2 : versions 1.24.17, 1.25.14, 1.26.9 avec Rancher Manager 2.7.9
Plateforme de conteneurs Red Hat OpenShift	4.12 à 4.14

Ces instructions supposent que vous avez déjà créé un cluster privé et préparé une méthode sécurisée pour y accéder à distance.

Pour ajouter votre cluster privé à Astra Control Service, vous devez effectuer les tâches suivantes :

1. Poser le connecteur Astra
2. Configuration du stockage persistant
3. Ajoutez le cluster privé autogéré à Astra Control Service

Poser le connecteur Astra

Avant d'ajouter un cluster privé, vous devez installer Astra Connector sur le cluster afin qu'Astra Control puisse communiquer avec lui. Reportez-vous à la section "Installez la version précédente d'Astra Connector pour les clusters privés gérés avec des workflows non natifs Kubernetes" pour obtenir des instructions.

Configuration du stockage persistant

Configurer le stockage persistant pour le cluster. Pour plus d'informations sur la configuration du stockage persistant, reportez-vous à la documentation de mise en route :

- "Configuration de Microsoft Azure avec Azure NetApp Files"
- "Configuration de Microsoft Azure avec des disques gérés Azure"
- "Configurer Amazon Web Services"
- "Configurez Google Cloud"

Ajoutez le cluster privé autogéré à Astra Control Service

Vous pouvez maintenant ajouter le cluster privé à Astra Control Service.

Avant de commencer

Un cluster autogéré est un cluster que vous provisionnez et gérez directement. ASTRA Control Service prend en charge les clusters autogérés qui s'exécutent dans un environnement de cloud public. Vos clusters autogérés peuvent utiliser Astra Control Provisioner pour s'interfacer avec les services de stockage NetApp ou des pilotes Container Storage interface (CSI) pour s'interfacer avec Amazon Elastic Block Store (EBS), les disques gérés Azure et le service Google persistent Disk.

ASTRA Control Service prend en charge les clusters autogérés qui utilisent les distributions Kubernetes suivantes :

- Plateforme de conteneurs Red Hat OpenShift
- Moteur rancher Kubernetes
- Kubernetes en amont

Votre cluster autogéré doit répondre aux exigences suivantes :

- Le cluster doit être accessible via Internet.
- Si vous utilisez ou prévoyez d'utiliser le stockage activé avec des pilotes CSI, les pilotes CSI appropriés doivent être installés sur le cluster. Pour plus d'informations sur l'utilisation des pilotes CSI pour intégrer le stockage, reportez-vous à la documentation de votre service de stockage.
- Vous avez accès au fichier kubeconfig du cluster qui ne contient qu'un seul élément de contexte. Suivre "ces instructions" pour générer un fichier kubeconfig.
- Si vous ajoutez le cluster à l'aide d'un fichier kubeconfig qui fait référence à une autorité de certification privée (CA), ajoutez la ligne suivante au `cluster` section du fichier kubeconfig. Cela permet à Astra Control d'ajouter le cluster :

  ```
  insecure-skip-tls-verify: true
  ```

- **Rancher uniquement**: Lorsque vous gérez des clusters d'applications dans un environnement Rancher, modifiez le contexte par défaut du cluster d'applications dans le fichier kubeconfig fourni par Rancher pour utiliser un contexte de plan de contrôle au lieu du contexte du serveur d'API Rancher. La charge est réduite sur le serveur API Rancher et les performances sont améliorées.
- **Exigences du mécanisme de provisionnement Astra Control** : vous devez avoir un mécanisme de provisionnement Astra Control correctement configuré, y compris ses composants Astra Trident, pour gérer les clusters.
  - **Revoir les exigences de l'environnement Astra Trident** : avant d'installer ou de mettre à niveau Astra Control Provisioner, consultez le "systèmes front-end, systèmes back-end et configurations hôte pris en charge".
  - **Activer la fonctionnalité Astra Control Provisioner** : il est fortement recommandé d'installer Astra Trident 23.10 ou version ultérieure et de l'activer "Fonctionnalité de stockage avancée Astra Control Provisioner". Dans les prochaines versions, Astra Control ne prendra pas en charge Astra Trident si le mécanisme de provisionnement Astra Control n'est pas également activé.
  - **Configurer un back-end de stockage** : au moins un back-end de stockage doit l'être "Configuré dans Astra Trident" sur le cluster.
  - **Configurer une classe de stockage** : au moins une classe de stockage doit être "Configuré dans Astra Trident" sur le cluster. Si une classe de stockage par défaut est configurée, assurez-vous qu'il s'agit de la classe de stockage **Only** qui possède l'annotation par défaut.

- **Configurer un contrôleur de snapshot de volume et installer une classe de snapshot de volume** : "Installez un contrôleur de snapshot de volume" Il est ainsi possible de créer des snapshots dans Astra Control. "Création" au moins un `VolumeSnapshotClass` Avec Astra Trident.

Étapes

1. Dans le Tableau de bord, sélectionnez **Manage Kubernetes cluster**.

   Suivez les invites pour ajouter le cluster.

2. **Fournisseur** : sélectionnez l'onglet **autre** pour ajouter des détails sur votre cluster auto-géré.
3. **Autre**: Fournir des détails sur votre cluster auto-géré en téléchargeant un `kubeconfig.yaml` ou en collant le contenu du `kubeconfig.yaml` fichier à partir du presse-papiers.

   Si vous créez la vôtre `kubeconfig` fichier, vous ne devez définir que **un** élément de contexte dans celui-ci. Reportez-vous à la section "ces instructions" pour plus d'informations sur la création `kubeconfig` fichiers.

4. **Nom d'identification** : indiquez un nom pour les informations d'identification de cluster autogérées que vous téléchargez sur Astra Control. Par défaut, le nom des identifiants est automatiquement renseigné comme nom du cluster.
5. **Identificateur de route privée** : saisissez l'identificateur de route privée que vous pouvez obtenir à partir du connecteur Astra. Si vous interrogez le connecteur Astra via le `kubectl get astraconnector -n astra-connector` l'identificateur de route privée est appelé `ASTRACONNECTORID`.

   L'identifiant de la route privée est le nom associé à Astra Connector qui permet de gérer un cluster Kubernetes privé par Astra. Dans ce contexte, un cluster privé est un cluster Kubernetes qui n'expose pas son serveur d'API à Internet.

6. Sélectionnez **Suivant**.
7. (Facultatif) **Storage** : si vous le souhaitez, sélectionnez la classe de stockage que les applications Kubernetes déployées sur ce cluster doivent utiliser par défaut.
   a. Pour sélectionner une nouvelle classe de stockage par défaut pour le cluster, cochez la case **affecter une nouvelle classe de stockage par défaut**.
   b. Sélectionnez une nouvelle classe de stockage par défaut dans la liste.

Chaque fournisseur de service de stockage cloud affiche les informations suivantes en matière de prix, de performance et de résilience :

- Cloud Volumes Service pour Google Cloud : informations sur le prix, la performance et la résilience
- Google persistent Disk : pas d'informations sur le prix, la performance ou la résilience disponibles
- Azure NetApp Files : informations sur les performances et la résilience
- Azure Managed Disks : aucun prix, performances ou résilience disponibles
- Amazon Elastic Block Store : pas d'informations disponibles sur le prix, la performance ou la résilience
- Amazon FSX pour NetApp ONTAP : aucune information disponible concernant le prix, les performances ou la résilience
- NetApp Cloud Volumes ONTAP : aucune information disponible sur le prix, les performances ou la résilience

Chaque classe de stockage peut utiliser l'un des services suivants :

- "Cloud Volumes Service pour Google Cloud"
- "Disque persistant Google"
- "Azure NetApp Files"
- "Disques gérés Azure"
- "Amazon Elastic Block Store"
- "Amazon FSX pour NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

En savoir plus sur "Classes de stockage pour les clusters Amazon Web Services". En savoir plus sur "Classes de stockage pour les clusters AKS". En savoir plus sur "Classes de stockage pour clusters GKE".

c. Sélectionnez **Suivant**.
d. **Revoir et approuver** : consultez les détails de la configuration.
e. Sélectionnez **Ajouter** pour ajouter le cluster à Astra Control Service.

Modifiez la classe de stockage par défaut

Vous pouvez modifier la classe de stockage par défaut d'un cluster.

Modifiez la classe de stockage par défaut avec Astra Control

Vous pouvez modifier la classe de stockage par défaut d'un cluster depuis Astra Control. Si votre cluster utilise un service back-end de stockage installé précédemment, il se peut que vous ne puissiez pas utiliser cette méthode pour modifier la classe de stockage par défaut (l'action **Set as default** n'est pas sélectionnable). Dans ce cas, vous pouvez Modifiez la classe de stockage par défaut à l'aide de la ligne de commande.

Étapes

1. Dans l'interface utilisateur du service de contrôle Astra, sélectionnez **clusters**.

2. Sur la page **clusters**, sélectionnez le cluster que vous souhaitez modifier.
3. Sélectionnez l'onglet **stockage**.
4. Sélectionnez la catégorie **classes de stockage**.
5. Sélectionnez le menu **actions** pour la classe de stockage que vous souhaitez définir par défaut.
6. Sélectionnez **définir comme valeur par défaut**.

Modifiez la classe de stockage par défaut à l'aide de la ligne de commande

Vous pouvez modifier la classe de stockage par défaut d'un cluster à l'aide des commandes Kubernetes. Cette méthode fonctionne quelle que soit la configuration du cluster.

Étapes

1. Connectez-vous à votre cluster Kubernetes.
2. Lister les classes de stockage de votre cluster :

```
kubectl get storageclass
```

3. Supprimez la désignation par défaut de la classe de stockage par défaut. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Sélectionnez par défaut une classe de stockage différente. Remplacez <SC_NAME> par le nom de la classe de stockage :

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirmez la nouvelle classe de stockage par défaut :

```
kubectl get storageclass
```

Vérifiez la version d'Astra Trident

Pour ajouter un cluster autogéré qui utilise Astra Control Provisioner ou Astra Trident pour les services de stockage, assurez-vous que la version installée d'Astra Trident est la version 23.10 ou la plus récente.

Étapes

1. Déterminez la version d'Astra Trident que vous exécutez :

```
kubectl get tridentversions -n trident
```

Si Astra Trident est installé, le résultat est similaire à ce qui suit :

```
NAME      VERSION
trident   24.02.0
```

Si Astra Trident n'est pas installé, le résultat est similaire à ce qui suit :

```
error: the server doesn't have a resource type "tridentversions"
```

2. Effectuez l'une des opérations suivantes :
   - Si vous exécutez Astra Trident 23.01 ou une version antérieure, utilisez-les "instructions" Pour effectuer une mise à niveau vers une version plus récente d'Astra Trident avant de passer à Astra Control Provisioner. C'est possible "effectuer une mise à niveau directe" Vers Astra Control Provisioner 24.02 si votre Astra Trident se trouve dans une fenêtre à quatre versions de la version 24.02. Par exemple, vous pouvez effectuer une mise à niveau directe d'Astra Trident 23.04 vers Astra Control Provisioner 24.02.
   - Si vous exécutez Astra Trident 23.10 ou version ultérieure, vérifiez que le mécanisme de provisionnement Astra Control a été utilisé "activé". ASTRA Control Provisioner ne fonctionnera pas avec les versions d'Astra Control Center antérieures à 23.10. "Mettez à niveau votre mécanisme de provisionnement Astra Control" De sorte qu'il dispose de la même version que l'Astra Control Center que vous mettez à niveau pour accéder aux dernières fonctionnalités.
3. Assurez-vous que les pods fonctionnent :

```
kubectl get pods -n trident
```

4. Vérifiez si les classes de stockage utilisent les pilotes Trident Astra pris en charge. Le nom de provisionnement doit être `csi.trident.netapp.io`. Reportez-vous à l'exemple suivant :

```
kubectl get sc
```

Exemple de réponse :

```
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE    ALLOWVOLUMEEXPANSION    AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate            true                    5d23h
```

Créez un fichier kubeconfig

Vous pouvez ajouter un cluster à Astra Control Service à l'aide d'un fichier kubeconfig. Selon le type de cluster à ajouter, vous devrez peut-être créer manuellement un fichier kubeconfig pour votre cluster en suivant des étapes spécifiques.

- Créez un fichier kubeconfig pour les clusters Amazon EKS
- Créez un fichier kubeconfig pour les clusters Red Hat OpenShift Service sur AWS (ROSA)
- Créez un fichier kubeconfig pour d'autres types de clusters

Créez un fichier kubeconfig pour les clusters Amazon EKS

Suivez ces instructions pour créer un fichier kubeconfig et un code secret de jeton permanent pour les clusters Amazon EKS. Un code secret de jeton permanent est requis pour les clusters hébergés dans EKS.

Étapes

1. Suivez les instructions de la documentation Amazon pour générer un fichier kubeconfig :

   "Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS"

2. Créer un compte de service comme suit :

   a. Créez un fichier de compte de service appelé `astracontrol-service-account.yaml`.

      Ajustez le nom du compte de service si nécessaire. L'espace de noms `kube-system` est nécessaire pour ces étapes. Si vous modifiez le nom du compte de service ici, vous devez appliquer les mêmes modifications dans les étapes suivantes.

```
<strong>astracontrol-service-account.yaml</strong>
```

   +

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. Appliquer le compte de service :

```
kubectl apply -f astracontrol-service-account.yaml
```

4. Créer un `ClusterRoleBinding` fichier appelé `astracontrol-clusterrolebinding.yaml`.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

5. Appliquer la liaison de rôle de cluster :

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

6. Créez un fichier secret de token de compte de service appelé `astracontrol-secret.yaml`.

```
<strong>astracontrol-secret.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. Appliquer le secret de jeton :

```
kubectl apply -f astracontrol-secret.yaml
```

8. Récupérer le secret de jeton :

```
kubectl get secret astra-admin-account -n kube-system -o
jsonpath='{.data.token}' | base64 -d
```

9. Remplacer l' `user` Section du fichier kubeconfig AWS EKS avec le jeton, comme illustré ci-dessous :

```
user:
    token: k8s-aws-
v1.aHR0cHM6Ly9zdHMudXMtd2VzdC0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ
XJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQ
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBM1JEWDdKU0haWU9LSEQ2SyUyRjIwMjMwN
DAzJTJGdXMtd2VzdC0yJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzA0M
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQ
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YjU4ZWM0NzdiM2NkZGYxNGRhNzU4MGI2Z
WQ2zY2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

Créez un fichier kubeconfig pour les clusters Red Hat OpenShift Service sur AWS (ROSA)

Suivez ces instructions pour créer un fichier kubeconfig pour les clusters Red Hat OpenShift Service sur AWS (ROSA).

Étapes

1. Connectez-vous au cluster ROSA.
2. Créer un compte de service :

```
oc create sa astracontrol-service-account
```

3. Ajouter un rôle de cluster :

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-
service-account
```

4. À l'aide de l'exemple suivant, créez un fichier de configuration secret de compte de service :

```
<strong>secret-astra-sa.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-account"
type: kubernetes.io/service-account-token
```

5. Créez le secret :

```
oc create -f secret-astra-sa.yaml
```

6. Modifiez le compte de service que vous avez créé et ajoutez le nom secret du compte de service Astra Control au `secrets` section :

```
oc edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-dvfcd
kind: ServiceAccount
metadata:
  creationTimestamp: "2023-08-04T04:18:30Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "169770"
  uid: 965fa151-923f-4fbd-9289-30cad15998ac
secrets:
- name: astracontrol-service-account-dockercfg-dvfcd
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```

7. Indiquez les secrets du compte de service, en les remplaçant `<CONTEXT>` avec le contexte approprié pour votre installation :

```
kubectl get serviceaccount astracontrol-service-account --context
<CONTEXT> --namespace default -o json
```

La fin de la sortie doit ressembler à ce qui suit :

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-dvfcd"},
{ "name": "secret-astracontrol-service-account"}
]
```

Les indices pour chaque élément dans `secrets` la matrice commence par 0. Dans l'exemple ci-dessus, l'index de `astracontrol-service-account-dockercfg-dvfcd` serait 0 et l'index pour `secret-astracontrol-service-account` serait 1. Dans votre sortie, notez le numéro d'index du compte de service secret. Vous aurez besoin de ce numéro d'index à l'étape suivante.

8. Générez le kubeconfig comme suit :

   a. Créer un `create-kubeconfig.sh` fichier. Remplacement `TOKEN_INDEX` au début du script suivant avec la valeur correcte.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
```

```
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. Source des commandes à appliquer à votre cluster Kubernetes.

```
source create-kubeconfig.sh
```

9. (Facultatif) Renommer le kubeconfig pour nommer votre cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Créez un fichier kubeconfig pour d'autres types de clusters

Suivez ces instructions pour créer un fichier kubeconfig de rôle limité ou étendu pour les clusters Rancher, Kubernetes en amont et Red Hat OpenShift.

Pour les clusters gérés à l'aide de kubeconfig, vous pouvez éventuellement créer une autorisation limitée ou un rôle d'administrateur d'autorisations étendues pour Astra Control Service.

Cette procédure vous aide à créer un kubeconfig distinct si l'un des scénarios suivants s'applique à votre environnement :

- Vous souhaitez limiter les autorisations Astra Control sur les clusters qu'il gère
- Vous utilisez plusieurs contextes et ne pouvez pas utiliser le kubeconfig Astra Control par défaut configuré lors de l'installation, sinon un rôle limité avec un seul contexte ne fonctionnera pas dans votre environnement

Avant de commencer

Assurez-vous que vous disposez des éléments suivants pour le cluster que vous souhaitez gérer avant

d'effectuer la procédure suivante :

- A "version prise en charge" de kubectl est installé.
- Kubectl accès au cluster que vous envisagez d'ajouter et de gérer avec Astra Control Service

> Pour cette procédure, vous n'avez pas besoin d'un accès kubectl au cluster exécutant Astra Control Service.

- Un kubeconfig actif pour le cluster que vous avez l'intention de gérer avec des droits d'administrateur de cluster pour le contexte actif

Étapes

1. Créer un compte de service :
   a. Créez un fichier de compte de service appelé `astracontrol-service-account.yaml`.

```
<strong>astracontrol-service-account.yaml</strong>
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

   b. Appliquer le compte de service :

```
kubectl apply -f astracontrol-service-account.yaml
```

2. Créez l'un des rôles de cluster suivants avec des autorisations suffisantes pour qu'un cluster soit géré par Astra Control :

Rôle limité du cluster

Ce rôle contient les autorisations minimales nécessaires à la gestion d'un cluster par Astra Control :

a. Créer un `ClusterRole` fichier appelé, par exemple, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```
  - horizontalpodautoscalers
  - ingresses
  - jobs
  - namespaces
  - networkpolicies
  - persistentvolumeclaims
  - poddisruptionbudgets
  - pods
  - podtemplates
  - replicasets
  - replicationcontrollers
  - replicationcontrollers/scale
  - rolebindings
  - roles
  - secrets
  - serviceaccounts
  - services
  - statefulsets
  - tridentmirrorrelationships
  - tridentsnapshotinfos
  - volumesnapshots
  - volumesnapshotcontents
  verbs:
  - delete

# Watch resources
# Necessary to monitor progress
- apiGroups:
  - ""
  resources:
  - pods
  - replicationcontrollers
  - replicationcontrollers/scale
  verbs:
  - watch

# Update resources
- apiGroups:
  - ""
  - build.openshift.io
  - image.openshift.io
  resources:
  - builds/details
  - replicationcontrollers
  - replicationcontrollers/scale
  - imagestreams/layers
```

```
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Pour les clusters OpenShift uniquement) Ajouter les éléments suivants à la fin du `astra-admin-account.yaml` fichier :

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. Appliquer le rôle de cluster :

```
kubectl apply -f astra-admin-account.yaml
```

Rôle de cluster étendu

Ce rôle contient des autorisations étendues pour qu'un cluster soit géré par Astra Control. Vous pouvez utiliser ce rôle si vous utilisez plusieurs contextes et que vous ne pouvez pas utiliser le kubeconfig Astra Control par défaut configuré lors de l'installation, ou si un rôle limité avec un seul contexte ne fonctionnera pas dans votre environnement :

Les éléments suivants `ClusterRole` Les étapes constituent un exemple Kubernetes général. Pour des instructions spécifiques à votre environnement, reportez-vous à la documentation de votre distribution Kubernetes.

a. Créer un `ClusterRole` fichier appelé, par exemple, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. Appliquer le rôle de cluster :

```
kubectl apply -f astra-admin-account.yaml
```

3. Créer la liaison de rôle cluster pour le rôle cluster vers le compte de service :

a. Créer un `ClusterRoleBinding` fichier appelé `astracontrol-clusterrolebinding.yaml`.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

b. Appliquer la liaison de rôle de cluster :

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. Créez et appliquez le secret de jeton :
   a. Créez un fichier secret de jeton appelé `secret-astracontrol-service-account.yaml`.

```
<strong>secret-astracontrol-service-account.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. Appliquer le secret de jeton :

```
kubectl apply -f secret-astracontrol-service-account.yaml
```

5. Ajoutez le secret de jeton au compte de service en ajoutant son nom au `secrets` tableau (dernière ligne de l'exemple suivant) :

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. Indiquez les secrets du compte de service, en les remplaçant `<context>` avec le contexte approprié pour votre installation :

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

La fin de la sortie doit ressembler à ce qui suit :

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Les indices pour chaque élément dans `secrets` la matrice commence par 0. Dans l'exemple ci-dessus, l'index de `astracontrol-service-account-dockercfg-48xhx` serait 0 et l'index pour `secret-astracontrol-service-account` serait 1. Dans votre sortie, notez le numéro d'index du compte de service secret. Vous aurez besoin de ce numéro d'index à l'étape suivante.

7. Générez le kubeconfig comme suit :
   a. Créer un `create-kubeconfig.sh` fichier.
   b. Remplacement `TOKEN_INDEX` au début du script suivant avec la valeur correcte.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
```

```
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

c. Source des commandes à appliquer à votre cluster Kubernetes.

```
source create-kubeconfig.sh
```

8. (Facultatif) Renommer le kubeconfig pour nommer votre cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Informations sur le copyright

Copyright © 2024 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTUELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site http://www.netapp.com/TM sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.