Configurer le centre de contrôle Astra

Astra Control Center

NetApp
April 25, 2024

Sommaire

Configurer le centre de contrôle Astra

Ajoutez une licence pour Astra Control Center

Lorsque vous installez Astra Control Center, une licence d'évaluation intégrée est déjà installée. Si vous évaluez Astra Control Center, vous pouvez ignorer cette étape.

Vous pouvez ajouter une nouvelle licence à l'aide de l'interface utilisateur Astra Control ou "API de contrôle Astra".

Les licences Astra Control Center mesurent les ressources CPU avec des unités de processeur Kubernetes et représentent les ressources de processeur attribuées aux nœuds de travail de tous les clusters Kubernetes gérés. Les licences dépendent de l'utilisation des processeurs virtuels. Pour plus d'informations sur le calcul des licences, reportez-vous à la section "Licences".

> Si votre installation dépasse le nombre de processeurs sous licence, Astra Control Center vous empêche de gérer de nouvelles applications. Une alerte s'affiche lorsque la capacité est dépassée.

> Pour mettre à jour une évaluation existante ou une licence complète, reportez-vous à la section "Mettre à jour une licence existante".

Avant de commencer

- Accès à une instance Astra Control Center récemment installée.
- Autorisations de rôle d'administrateur.
- A "Fichier de licence NetApp" (NLF).

Étapes

1. Connectez-vous à l'interface utilisateur du centre de contrôle Astra.
2. Sélectionnez **compte** > **Licence**.
3. Sélectionnez **Ajouter licence**.
4. Accédez au fichier de licence (NLF) que vous avez téléchargé.
5. Sélectionnez **Ajouter licence**.

La page **Account** > **License** affiche les informations de licence, la date d'expiration, le numéro de série de licence, l'ID de compte et les unités UC utilisées.

> Si vous disposez d'une licence d'évaluation et que vous n'envoyez pas de données à AutoSupport, assurez-vous de stocker votre identifiant de compte pour éviter toute perte de données en cas de défaillance d'Astra Control Center.

Activez le mécanisme de provisionnement Astra Control

Les versions 23.10 et ultérieures d'Astra Trident incluent la possibilité d'utiliser Astra Control Provisioner qui permet aux utilisateurs d'Astra Control sous licence d'accéder à des fonctionnalités avancées de provisionnement du stockage. ASTRA Control

Provisioner offre cette fonctionnalité étendue en plus des fonctionnalités standard d'Astra Trident CSI.

Dans les prochaines mises à jour d'Astra Control, Astra Control Provisioner remplacera Astra Trident en tant que mécanisme de provisionnement et d'orchestration du stockage, et sera obligatoire pour l'utilisation d'Astra Control. Pour cette raison, il est fortement recommandé aux utilisateurs d'Astra Control d'activer Astra Control Provisioner. ASTRA Trident continuera à rester open source et sera publié, maintenu, pris en charge et mis à jour avec le nouveau CSI et d'autres fonctionnalités de NetApp.

Description de la tâche

Vous devez suivre cette procédure si vous êtes un utilisateur d'Astra Control Center sous licence et que vous cherchez à utiliser la fonctionnalité Astra Control Provisioner. Vous devez également suivre cette procédure si vous êtes un utilisateur d'Astra Trident et souhaitez utiliser les fonctionnalités supplémentaires d'Astra Control Provisioner sans également utiliser Astra Control.

Pour chaque cas, la fonctionnalité de provisionneur n'est pas activée par défaut dans Astra Trident 24.02 et doit être activée.

Avant de commencer

Si vous activez Astra Control Provisioner, effectuez d'abord les opérations suivantes :

ASTRA Control assure aux utilisateurs un provisionnement avec Astra Control Center

- **Obtenir une licence Astra Control Center** : vous aurez besoin d'une licence "Licence Astra Control Center" Pour activer le mécanisme de provisionnement Astra Control et accéder aux fonctionnalités qu'il fournit.
- **Installer ou mettre à niveau vers Astra Control Center 23.10 ou version ultérieure**: Vous aurez besoin de la dernière version d'Astra Control Center (24.02) si vous prévoyez d'utiliser la dernière fonctionnalité d'Astra Control Provisioner (24.02) avec Astra Control.
- **Confirmez que votre cluster a une architecture système AMD64** : l'image Astra Control Provisioner est fournie dans les architectures CPU AMD64 et ARM64, mais seul AMD64 est pris en charge par Astra Control Center.
- **Obtenez un compte Astra Control Service pour l'accès au registre**: Si vous avez l'intention d'utiliser le registre Astra Control plutôt que le site de support NetApp pour télécharger l'image Astra Control provisionner, effectuez l'enregistrement pour un "Compte Astra Control Service". Une fois que vous aurez rempli le formulaire, envoyé son formulaire et créé un compte BlueXP, vous recevrez un e-mail de bienvenue Astra Control Service.
- **Si vous avez installé Astra Trident, vérifiez que sa version se trouve dans une fenêtre à quatre versions**: Vous pouvez effectuer une mise à niveau directe vers Astra Trident 24.02 avec Astra Control Provisioner 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 la version 24.02.

Utilisateurs d'Astra Control Provisioner uniquement

- **Obtenir une licence Astra Control Center** : vous aurez besoin d'une licence "Licence Astra Control Center" Pour activer le mécanisme de provisionnement Astra Control et accéder aux fonctionnalités qu'il fournit.
- **Si vous avez installé Astra Trident, vérifiez que sa version se trouve dans une fenêtre à quatre versions**: Vous pouvez effectuer une mise à niveau directe vers Astra Trident 24.02 avec Astra Control Provisioner 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 la version 24.02.
- **Obtenez un compte Astra Control Service pour l'accès au registre**: Vous aurez besoin d'accéder au registre pour télécharger les images d'Astra Control provisionner. Pour commencer, terminez l'inscription à un "Compte Astra Control Service". Une fois que vous aurez rempli le formulaire, envoyé son formulaire et créé un compte BlueXP, vous recevrez un e-mail de bienvenue Astra Control Service.

(Étape 1) Obtenez l'image Astra Control Provisioner

Les utilisateurs d'Astra Control Center peuvent obtenir l'image d'Astra Control provisionner en utilisant le registre Astra Control ou la méthode du site de support NetApp. Les utilisateurs d'Astra Trident qui souhaitent utiliser Astra Control Provisioner sans Astra Control doivent utiliser la méthode de Registre.

Registre d'images Astra Control

> Vous pouvez utiliser Podman à la place de Docker pour les commandes de cette procédure. Si vous utilisez un environnement Windows, PowerShell est recommandé.

1. Accédez au registre d'images NetApp Astra Control :
   a. Connectez-vous à l'interface utilisateur Web d'Astra Control Service et sélectionnez l'icône de figure en haut à droite de la page.
   b. Sélectionnez **accès API**.
   c. Notez votre ID de compte.
   d. A partir de la même page, sélectionnez **générer jeton API** et copiez la chaîne de jeton API dans le presse-papiers et enregistrez-la dans votre éditeur.
   e. Connectez-vous au registre Astra Control à l'aide de la méthode de votre choix :

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

```
crane auth login cr.astra.netapp.io -u <account-id> -p <api-
token>
```

2. (Registres personnalisés uniquement) Suivez ces étapes pour déplacer l'image vers votre registre personnalisé. Si vous n'utilisez pas de registre, suivez les étapes de l'opérateur Trident dans le "section suivante".
   a. Extrayez l'image Astra Control Provisioner du Registre :

> L'image extraite ne prend pas en charge plusieurs plates-formes et ne prend en charge que la même plate-forme que l'hôte qui a extrait l'image, comme Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Exemple :

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform
linux/amd64
```

   a. Marquer l'image :

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

b. Envoyez l'image vers votre registre personnalisé :

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

> Vous pouvez utiliser Crane Copy comme alternative à l'exécution des commandes Docker suivantes :
> `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

Site de support NetApp

1. Téléchargez le bundle Astra Control Provisioner (`trident-acp-[version].tar`) du "Page de téléchargements d'Astra Control Center".
2. (Recommandé mais facultatif) Téléchargez le bundle de certificats et de signatures pour Astra Control Center (astra-control-Center-certs-[version].tar.gz) pour vérifier la signature du bundle trident-acp-[version] tar.

```
tar -vxzf astra-control-center-certs-[version].tar.gz
```

```
openssl dgst -sha256 -verify certs/AstraControlCenterDockerImages-public.pub -signature certs/trident-acp-[version].tar.sig trident-acp-[version].tar
```

3. Charger l'image Astra Control Provisioner :

```
docker load < trident-acp-24.02.0.tar
```

Réponse :

```
Loaded image: trident-acp:24.02.0-linux-amd64
```

4. Marquer l'image :

```
docker tag trident-acp:24.02.0-linux-amd64 <my_custom_registry>/trident-acp:24.02.0
```

5. Envoyez l'image vers votre registre personnalisé :

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

(Étape 2) Activer le provisionnement Astra Control dans Astra Trident

Déterminez si la méthode d'installation d'origine utilisait un "Opérateur (manuellement ou avec Helm) ou tridentctl" et suivez les étapes appropriées selon votre méthode d'origine.

Opérateur Astra Trident

1. "Téléchargez le programme d'installation d'Astra Trident et extrayez-le".
2. Si vous n'avez pas encore installé Astra Trident ou si vous avez supprimé l'opérateur de votre déploiement Astra Trident d'origine :
   a. Créez le CRD :

   ```
   kubectl create -f
   deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.y
   aml
   ```

   b. Créer l'espace de nom trident (`kubectl create namespace trident`) ou confirmez que l'espace de nom trident existe toujours (`kubectl get all -n trident`). Si l'espace de noms a été supprimé, créez-le à nouveau.
3. Mettez à jour Astra Trident vers la version 24.02.0 :

   > Pour les clusters exécutant Kubernetes 1.24 ou version antérieure, utilisez `bundle_pre_1_25.yaml`. Pour les clusters exécutant Kubernetes 1.25 ou version ultérieure, utilisez `bundle_post_1_25.yaml`.

   ```
   kubectl -n trident apply -f trident-installer/deploy/<bundle-
   name.yaml>
   ```

4. Vérifiez que Astra Trident est en cours d'exécution :

   ```
   kubectl get torc -n trident
   ```

   Réponse :

   ```
   NAME      AGE
   trident   21m
   ```

5. si vous avez un registre qui utilise des secrets, créez un secret à utiliser pour extraire l'image Astra Control Provisioner :

   ```
   kubectl create secret docker-registry <secret_name> -n trident
   --docker-server=<my_custom_registry> --docker-username=<username>
   --docker-password=<token>
   ```

6. Modifiez la CR TridentOrchestrator et apportez les modifications suivantes :

```
kubectl edit torc trident -n trident
```

a. Définissez un emplacement de Registre personnalisé pour l'image Astra Trident ou extrayez-le du Registre Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0` ou `tridentImage: netapp/trident:24.02.0`).

b. Activez le mécanisme de provisionnement Astra Control (`enableACP: true`).

c. Définissez l'emplacement de registre personnalisé pour l'image Astra Control Provisioner ou extrayez-le du registre Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0` ou `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Si vous avez établi secrets d'extraction d'image plus tôt dans cette procédure, vous pouvez les définir ici (`imagePullSecrets: - <secret_name>`). Utilisez le même nom secret que celui que vous avez établi lors des étapes précédentes.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
  - <secret_name>
```

7. Enregistrez et quittez le fichier. Le processus de déploiement commence automatiquement.
8. Vérifiez que l'opérateur, le déploiement et les réplicateurs sont créés.

```
kubectl get all -n trident
```

> Il ne doit y avoir que **une instance** de l'opérateur dans un cluster Kubernetes. Ne créez pas plusieurs déploiements de l'opérateur Astra Trident.

9. Vérifiez le `trident-acp` le conteneur est en cours d'exécution `acpVersion` est `24.02.0` avec un état de `Installed`:

```
kubectl get torc -o yaml
```

Réponse :

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

tridentctl

1. "Téléchargez le programme d'installation d'Astra Trident et extrayez-le".
2. "Si vous disposez d'une Astra Trident, désinstallez-la du cluster qui l'héberge".
3. Installez Astra Trident avec Astra Control Provisioner activé (`--enable-acp=true`) :

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Vérifiez que le mécanisme de provisionnement Astra Control a été activé :

```
./tridentctl -n trident version
```

Réponse :

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 24.02.0 | 24.02.0 | 24.02.0. | +----------------
+----------------+-------------+
```

Gouvernail

1. Si vous avez installé Astra Trident 23.07.1 ou une version antérieure, "désinstaller" l'opérateur et les autres composants.
2. Si votre cluster Kubernetes s'exécute sur la version 1.24 ou antérieure, supprimez la psp :

```
kubectl delete psp tridentoperatorpod
```

3. Ajout du référentiel Astra Trident Helm :

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

4. Mettre à jour le graphique Helm :

```
helm repo update netapp-trident
```

Réponse :

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. □Happy Helming!□
```

5. Répertorier les images :

```
./tridentctl images -n trident
```

Réponse :

```
| v1.28.0            | netapp/trident:24.02.0|
|                    | docker.io/netapp/trident-autosupport:24.02|
|                    | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                    | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                    | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                    | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                    | registry.k8s.io/sig-storage/csi-node-driver-
registrar:v2.10.0 |
|                    | netapp/trident-operator:24.02.0 (optional)
```

6. Vérifier que trident-Operator 24.02.0 est disponible :

```
helm search repo netapp-trident/trident-operator --versions
```

Réponse :

```
NAME                            CHART VERSION   APP VERSION
DESCRIPTION
netapp-trident/trident-operator 100.2402.0      24.02.0         A
```

7. Utiliser `helm install` et exécutez l'une des options suivantes qui incluent ces paramètres :
   - Un nom pour votre emplacement de déploiement
   - Version d'Astra Trident
   - Nom de l'image Astra Control Provisioner
   - Indicateur d'activation du provisionneur
   - (Facultatif) Un chemin de registre local. Si vous utilisez un registre local, votre "Images Trident" Peut être situé dans un registre ou dans des registres différents, mais toutes les images CSI doivent se trouver dans le même registre.
   - Espace de noms Trident

Options

- Images sans registre

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0
--set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Images dans un ou plusieurs registres

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

Vous pouvez utiliser `helm list` pour vérifier les détails de l'installation tels que le nom, l'espace de noms, le graphique, l'état, la version de l'application, et numéro de révision.

Si vous rencontrez des problèmes pour déployer Trident à l'aide d'Helm, exécutez cette commande pour désinstaller complètement Astra Trident :

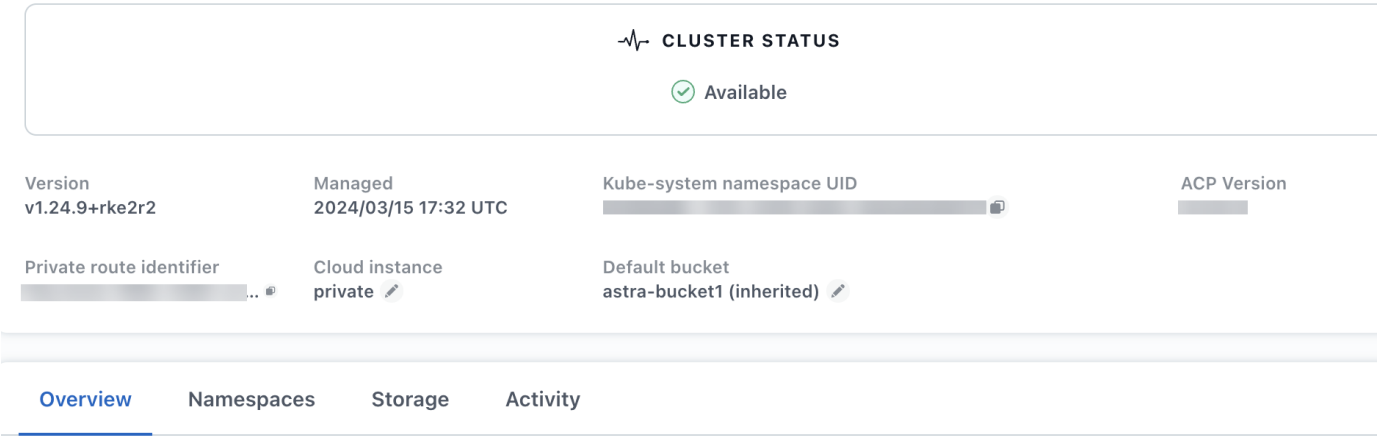
```
./tridentctl uninstall -n trident
```

Ne pas "Retirez complètement les CRD Astra Trident" Dans le cadre de votre désinstallation avant de tenter à nouveau d'activer Astra Control Provisioner.

Résultat

La fonctionnalité Astra Control Provisioner est activée et vous pouvez utiliser toutes les fonctions disponibles pour la version que vous exécutez.

(Pour les utilisateurs d'Astra Control Center uniquement) après l'installation d'Astra Control Provisioner, le cluster qui héberge le provisionneur dans l'interface utilisateur d'Astra Control Center affiche un `ACP version` plutôt que `Trident version` et le numéro de version actuellement installé.



Pour en savoir plus

- "Documentation sur les mises à niveau d'Astra Trident"

Préparez votre environnement à la gestion des clusters avec Astra Control

Avant d'ajouter un cluster, assurez-vous que les conditions préalables suivantes sont remplies. Vous devez également procéder à des vérifications d'éligibilité pour vous assurer que votre cluster est prêt à être ajouté à Astra Control Center et créer des rôles de cluster kubeconfig, si nécessaire.

ASTRA Control vous permet d'ajouter des clusters gérés par une ressource personnalisée (CR) ou un kubeconfig, en fonction de votre environnement et de vos préférences.

Avant de commencer

- **Respecter les conditions préalables environnementales** : votre environnement est conforme "de l'environnement opérationnel" Pour Astra Control Center.
- **Configurer les nœuds de travail** : assurez-vous que vous "configurez les nœuds worker" dans votre cluster avec les pilotes de stockage appropriés afin que les pods puissent interagir avec le stockage back-end.
- **Activer les restrictions PSA** : si l'application d'admission de sécurité du pod est activée sur votre cluster, ce qui est standard pour les clusters Kubernetes 1.25 et versions ultérieures, vous devez activer les restrictions PSA sur ces espaces de noms :

- `netapp-acc-operator` espace de noms :

```
kubectl label --overwrite ns netapp-acc-operator pod-
security.kubernetes.io/enforce=privileged
```

- `netapp monitoring` espace de noms :

```
kubectl label --overwrite ns netapp-monitoring pod-
security.kubernetes.io/enforce=privileged
```

- **Informations d'identification ONTAP** : vous avez besoin d'informations d'identification ONTAP et d'un superutilisateur et d'un ID utilisateur définis sur le système ONTAP de sauvegarde pour sauvegarder et restaurer des applications avec le Centre de contrôle Astra.

  Exécutez les commandes suivantes dans la ligne de commande ONTAP :

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **Configuration requise pour les clusters gérés par kubeconfig** : ces exigences sont spécifiques pour les clusters d'applications gérés par kubeconfig.
  - **Rendre kubeconfig. accessible**: Vous avez accès au "configuration par défaut du cluster" ça "vous avez configuré lors de l'installation".
  - **Considérations relatives à l'autorité de certification** : si vous ajoutez le cluster à l'aide d'un fichier kubeconfig qui fait référence à une autorité de certification privée (AC), 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.

Effectuer des vérifications d'éligibilité

Effectuez les contrôles d'éligibilité suivants pour vous assurer que votre grappe est prête à être ajoutée au Centre de contrôle Astra.

Étapes

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

   ```
   kubectl get tridentversion -n trident
   ```

   Si Astra Trident existe, le résultat de cette commande est similaire à ce qui suit :

   ```
   NAME      VERSION
   trident   24.02.0
   ```

   Si Astra Trident n'existe pas, 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 tous les modules (y compris `trident-acp`) sont en cours d'exécution :

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

4. Déterminez si les classes de stockage utilisent les pilotes Astra Trident pris en charge. Le nom de provisionnement doit être `csi.trident.netapp.io`. Voir 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 kubeconfig pour le rôle de cluster

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 Center. Il ne s'agit pas d'une procédure requise pour la configuration d'Astra Control Center, car vous avez déjà configuré un kubeconfig dans le cadre du "processus d'installation".

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 :

- kubectl v1.23 ou version ultérieure installée
- Accès kubectl au cluster que vous souhaitez ajouter et gérer avec Astra Control Center

> Pour cette procédure, il n'est pas nécessaire d'avoir un accès kubectl au cluster qui exécute Astra Control Center.

- 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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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
```

(Aperçu technique) installez Astra Connector pour les clusters gérés

Les clusters gérés par Astra Control Center utilisent Astra Connector pour faciliter la communication entre le cluster géré et Astra Control Center. Vous devez installer Astra Connector sur tous les clusters que vous souhaitez gérer.

Poser le connecteur Astra

Vous installez Astra Connector 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 Astra Control. 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
```

```yaml
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.

Ajouter un cluster

Pour commencer à gérer vos applications, ajoutez un cluster Kubernetes et gérez-le comme une ressource de calcul. Il faut ajouter un cluster pour découvrir vos applications Kubernetes pour Astra Control Center.

> Nous vous recommandons de gérer le cluster qu'Astra Control Center déploie en premier avant d'ajouter d'autres clusters à Astra Control Center. La gestion du cluster initial est nécessaire pour envoyer les données Kubemetrics et les données associées au cluster pour les mesures et le dépannage.

Avant de commencer

- Avant d'ajouter un cluster, vérifiez et effectuez les opérations nécessaires "tâches préalables".
- Si vous utilisez un pilote SAN ONTAP, assurez-vous que les chemins d'accès multiples sont activés sur tous vos clusters Kubernetes.

Étapes

1. Naviguer à partir du menu Tableau de bord ou clusters :
   - Dans **Dashboard**, sélectionnez **Add** dans le volet clusters.
   - Dans la zone de navigation de gauche, sélectionnez **clusters**, puis **Ajouter un cluster** à partir de la page clusters.
2. Dans la fenêtre **Ajouter un cluster** qui s'ouvre, chargez un `kubeconfig.yaml` classez le contenu d'un `kubeconfig.yaml` fichier.

> Le `kubeconfig.yaml` le fichier doit inclure **uniquement les informations d'identification du cluster pour un cluster**.

> 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. Si vous avez créé un kubeconfig pour un rôle de cluster limité à l'aide de "ce processus", assurez-vous de télécharger ou de coller ce kubeconfig dans cette étape.

3. Indiquez un nom d'identification. Par défaut, le nom des identifiants est automatiquement renseigné comme nom du cluster.
4. Sélectionnez **Suivant**.
5. Sélectionnez la classe de stockage par défaut à utiliser pour ce cluster Kubernetes et sélectionnez **Suivant**.

> Vous devez sélectionner une classe de stockage configurée dans Astra Control Provisioner et prise en charge par le stockage ONTAP.

6. Passez en revue les informations, et si tout semble bien, sélectionnez **Ajouter**.

Résultat

Le cluster passe à l'état **découverte**, puis passe à **sain**. Vous gérez maintenant le cluster avec Astra Control Center.

> Une fois que vous avez ajouté un cluster à gérer dans Astra Control Center, le déploiement de l'opérateur de surveillance peut prendre quelques minutes. En attendant, l'icône notification devient rouge et consigne un événement **échec de la vérification de l'état de l'agent de surveillance**. Vous pouvez ignorer cela car le problème résout lorsque le centre de contrôle Astra obtient le statut correct. Si le problème ne résout pas le problème en quelques minutes, accédez au cluster, puis exécutez-le `oc get pods -n netapp-monitoring` comme point de départ. Vous devrez consulter les journaux de l'opérateur de surveillance pour déboguer le problème.

Activez l'authentification sur un système back-end de stockage ONTAP

ASTRA Control Center offre deux modes d'authentification d'un backend ONTAP :

- **Authentification basée sur les informations d'identification** : le nom d'utilisateur et le mot de passe d'un utilisateur ONTAP avec les autorisations requises. Vous devez utiliser un rôle de connexion de sécurité prédéfini, tel que admin ou vsadmin, pour assurer une compatibilité maximale avec les versions de ONTAP.
- **Authentification basée sur un certificat** : Astra Control Center peut également communiquer avec un cluster ONTAP à l'aide d'un certificat installé sur le back-end. Vous devez utiliser le certificat client, la clé et le certificat de l'autorité de certification approuvée, le cas échéant (recommandé).

Vous pouvez par la suite mettre à jour les systèmes back-end existants pour passer d'un type d'authentification à une autre. Une seule méthode d'authentification est prise en charge à la fois.

Activer l'authentification basée sur les informations d'identification

ASTRA Control Center requiert les identifiants d'un cluster-scoped `admin` Pour communiquer avec le backend ONTAP. Vous devez utiliser des rôles standard prédéfinis, tels que `admin`. La compatibilité avec les futures versions d'ONTAP qui pourraient exposer les API de fonctionnalités à utiliser dans les futures versions d'Astra Control Center est ainsi garantie.

> Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Astra Control Center, mais il n'est pas recommandé.

Un exemple de définition de back-end se présente comme suit :

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

La définition du back-end est le seul endroit où les informations d'identification sont stockées en texte brut. La création ou la mise à jour d'un back-end est la seule étape qui nécessite la connaissance des informations d'identification. Il s'agit donc d'une opération réservée à l'administrateur du stockage ou de Kubernetes.

Activer l'authentification basée sur certificat

ASTRA Control Center peut utiliser des certificats pour communiquer avec les systèmes back-end ONTAP, nouveaux et existants. Vous devez entrer les informations suivantes dans la définition du back-end.

- `clientCertificate`: Certificat client.
- `clientPrivateKey`: Clé privée associée.
- `trustedCACertificate`: Certificat de l'autorité de certification approuvée. Si vous utilisez une autorité de certification approuvée, ce paramètre doit être fourni. Ceci peut être ignoré si aucune autorité de certification approuvée n'est utilisée.

Vous pouvez utiliser l'un des types de certificats suivants :

- Certificat auto-signé
- Certificat tiers

Activez l'authentification avec un certificat auto-signé

Un flux de travail type comprend les étapes suivantes.

Étapes

1. Générez un certificat client et une clé. Lors de la génération, définissez le nom commun (CN) sur l'utilisateur ONTAP pour s'authentifier en tant que.

   ```
   openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
   -out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
   ```

2. Installez le certificat client de type `client-ca` Et sur le cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Vérifiez que le rôle de connexion de sécurité ONTAP prend en charge la méthode d'authentification par certificat.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Tester l'authentification à l'aide du certificat généré. Remplacer <LIF> et <vserver name> de ONTAP par l'IP et le nom du SVM de la LIF de gestion. Vous devez vous assurer que le LIF a sa politique de service définie sur `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. À l'aide des valeurs obtenues à l'étape précédente, ajoutez le back-end de stockage dans l'interface utilisateur d'Astra Control Center.

Activez l'authentification à l'aide d'un certificat tiers

Si vous disposez d'un certificat tiers, vous pouvez configurer l'authentification basée sur un certificat à l'aide de ces étapes.

Étapes

1. Générer la clé privée et la RSC :

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>”
```

2. Transmettez la RSC à l'autorité de certification Windows (autorité de certification tierce) et émettez le certificat signé.
3. Téléchargez le certificat signé et nommez-le « ontap_signed_cert.crt ».
4. Exportez le certificat racine à partir de l'autorité de certification Windows (autorité de certification tierce).
5. Nommez ce fichier `ca_root.crt`

   Vous disposez maintenant des trois fichiers suivants :

- **Clé privée** : `ontap_signed_request.key` (Il s'agit de la clé correspondante pour le certificat de serveur dans ONTAP. Elle est nécessaire lors de l'installation du certificat du serveur.)
- **Certificat signé**: `ontap_signed_cert.crt` (Il s'agit également du *certificat de serveur* dans ONTAP.)
- **Certificat CA racine** : `ca_root.crt` (Il s'agit également du certificat *Server-ca* dans ONTAP.)

6. Installez ces certificats dans ONTAP. Générer et installer `server` et `server-ca` Certificats sur ONTAP.

Développez pour Sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Créez le certificat client pour le même hôte pour la communication sans mot de passe. ASTRA Control Center utilise ce processus pour communiquer avec ONTAP.
8. Générer et installer les certificats client sur ONTAP :

Développez pour Sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate's CA and serial number for reference:

CA:
serial:
The certificate's generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```

9. Ajoutez le système back-end de stockage dans l'interface utilisateur d'Astra Control Center et fournissez les valeurs suivantes :
   - **Certificat client** : ontap_test_client.pem
   - **Clé privée** : ontap_test_client.key
   - **Certificat CA de confiance** : ontap_signed_cert.crt

Ajout d'un système back-end

Après avoir configuré les informations d'identification ou d'authentification de certificat, vous pouvez ajouter un système back-end de stockage ONTAP existant à Astra Control Center pour gérer ses ressources.

La gestion des clusters de stockage d'Astra Control en tant que backend de stockage vous permet d'obtenir des liens entre les volumes persistants (PVS) et le back-end de stockage, ainsi que des metrics de stockage supplémentaires.

L'ajout et la gestion de systèmes back-end de stockage ONTAP dans Astra Control Center sont facultatifs si vous utilisez la technologie NetApp SnapMirror si vous avez activé Astra Control Provisioner.

Étapes

1. Dans la zone de navigation gauche du tableau de bord, sélectionnez **Backends**.
2. Sélectionnez **Ajouter**.
3. Dans la section utiliser existant de la page Ajouter un back-end de stockage, sélectionnez **ONTAP**.
4. Sélectionnez l'une des options suivantes :
   - **Utiliser les informations d'identification de l'administrateur** : saisissez l'adresse IP de gestion du cluster ONTAP et les informations d'identification de l'administrateur. Les identifiants doivent être identifiants au niveau du cluster.

     > L'utilisateur dont vous saisissez ici les informations d'identification doit disposer du `ontapi` Méthode d'accès de connexion utilisateur activée dans ONTAP System Manager sur le cluster ONTAP. Si vous prévoyez d'utiliser la réplication SnapMirror, appliquez les identifiants de l'utilisateur au rôle « admin », qui dispose des méthodes d'accès `ontapi` et `http`, Sur les clusters ONTAP source et destination. Reportez-vous à la section "Gérer les comptes utilisateur dans la documentation ONTAP" pour en savoir plus.

   - **Utiliser un certificat**: Télécharger le certificat `.pem` fichier, la clé de certificat `.key` et éventuellement le fichier de l'autorité de certification.
5. Sélectionnez **Suivant**.
6. Confirmez les détails du back-end et sélectionnez **gérer**.

Résultat

Le back-end s'affiche dans le `online` état dans la liste avec des informations récapitulatives.

> Vous devrez peut-être actualiser la page pour que le back-end apparaisse.

Ajouter un godet

Vous pouvez ajouter un compartiment à l'aide de l'interface utilisateur Astra Control ou "API de contrôle Astra". Il est essentiel d'ajouter des fournisseurs de compartiments de stockage objet pour sauvegarder les applications et le stockage persistant ou pour cloner les applications entre les clusters. Astra Control stocke les sauvegardes ou les clones dans les compartiments de magasin d'objets que vous définissez.

Si vous clonez la configuration de vos applications et le stockage persistant vers le même cluster, il n'est pas nécessaire d'utiliser un compartiment dans Astra Control. La fonctionnalité de copie Snapshot des applications ne nécessite pas de compartiment.

Avant de commencer

- Assurez-vous que vous disposez d'un compartiment accessible depuis vos clusters gérés par Astra Control Center.
- Vérifiez que vous disposez des informations d'identification pour le compartiment.
- S'assurer que le godet est de l'un des types suivants :

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Microsoft Azure
- S3 générique

> Amazon Web Services (AWS) et Google Cloud Platform (GCP) utilisent le type de compartiment S3 générique.

> Bien qu'Astra Control Center prenne en charge Amazon S3 en tant que fournisseur de compartiments génériques, Astra Control Center peut ne pas prendre en charge tous les fournisseurs de magasins d'objets qui affirment la prise en charge d'Amazon S3.

Étapes

1. Dans la zone de navigation de gauche, sélectionnez **godets**.
2. Sélectionnez **Ajouter**.
3. Sélectionner le type de godet.

> Lorsque vous ajoutez un compartiment, sélectionnez le fournisseur approprié et fournissez les identifiants appropriés pour ce fournisseur. Par exemple, l'interface utilisateur accepte NetApp ONTAP S3 comme type et accepte les identifiants StorageGRID. Toutefois, toutes les futures sauvegardes et restaurations des applications à l'aide de ce compartiment échoueront.

4. Saisissez un nom de compartiment existant et une description facultative.

> Le nom et la description du compartiment apparaissent comme un emplacement de sauvegarde que vous pouvez choisir plus tard lors de la création d'une sauvegarde. Ce nom apparaît également lors de la configuration de la règle de protection.

5. Entrez le nom ou l'adresse IP du terminal S3.
6. Sous **Sélectionner les informations d'identification**, choisissez l'onglet **Ajouter** ou **utiliser l'onglet existant**.
   - Si vous avez choisi **Ajouter**:
     i. Saisissez un nom pour l'identifiant qui le distingue des autres identifiants dans Astra Control.
     ii. Saisissez l'ID d'accès et la clé secrète en collant le contenu dans le presse-papiers.
   - Si vous avez choisi **utiliser existant**:
     i. Sélectionnez les informations d'identification existantes à utiliser avec le compartiment.
7. Sélectionnez `Add`.

> Lorsque vous ajoutez un godet, Astra Control marque un godet avec l'indicateur de compartiment par défaut. Le premier compartiment que vous créez devient le compartiment par défaut. Au fur et à mesure que vous ajoutez des compartiments, vous pourrez décider plus tard "définir un autre compartiment par défaut".