Configurer le centre de contrôle Astra

Astra Control Center

NetApp
November 21, 2023

Sommaire

Configurer le centre de contrôle Astra

Astra Control Center prend en charge et surveille ONTAP et Astra Data Store en tant que système back-end de stockage. Après avoir installé Astra Control Center, connectez-vous à l'interface utilisateur et modifiez votre mot de passe, vous devez configurer une licence, ajouter des clusters, gérer le stockage et ajouter des compartiments.

Tâches

- Ajoutez une licence pour Astra Control Center
- Ajouter un cluster
- Ajout d'un système back-end
- Ajouter un godet

Ajoutez une licence pour Astra Control Center

Vous pouvez ajouter une nouvelle licence à l'aide de l'interface utilisateur ou de "API" Pour bénéficier de toutes les fonctionnalités de l'Astra Control Center. Sans licence, votre utilisation d'Astra Control Center se limite à la gestion des utilisateurs et à l'ajout de nouveaux clusters.

Pour plus d'informations sur le calcul des licences, reportez-vous à la section "Licences".

> Pour mettre à jour une évaluation existante ou une licence complète, voir "Mettre à jour une licence existante".

Les licences Astra Control Center mesurent les ressources CPU avec des unités de processeur Kubernetes. La licence doit tenir compte des ressources CPU attribuées aux nœuds workers de tous les clusters Kubernetes gérés. Avant d'ajouter une licence, vous devez obtenir le fichier de licence (NLF) du "Site de support NetApp".

Vous pouvez également essayer Astra Control Center avec une licence d'évaluation qui vous permet d'utiliser Astra Control Center pendant 90 jours à compter de la date de téléchargement de la licence. Vous pouvez vous inscrire pour une version d'évaluation gratuite en vous inscrivant "ici".

> 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.

Ce dont vous avez besoin

Lorsque vous avez téléchargé Astra Control Center à partir du "Site de support NetApp", Vous avez également téléchargé le fichier de licence NetApp (NLF). Assurez-vous d'avoir accès à ce fichier de licence.

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, veillez à stocker votre identifiant de compte afin d'éviter toute perte de données en cas d'échec du Centre de contrôle Astra si vous n'envoyez pas d'ASUP.

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. Avec Astra Data Store, vous pouvez ajouter le cluster d'applications Kubernetes qui contient des applications qui utilisent des volumes provisionnés par Astra Data Store.

> 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. Vous pouvez utiliser la fonction **Ajouter un cluster** pour gérer un cluster avec Astra Control Center.

> Lorsque Astra Control gère un cluster, il conserve le suivi de la classe de stockage par défaut du cluster. Si vous modifiez la classe de stockage à l'aide de `kubectl` Contrôle Astra rétablit le changement. Pour modifier la classe de stockage par défaut d'un cluster géré par Astra Control, utilisez l'une des méthodes suivantes :
>
> - Utilisez l'API de contrôle Astra `PUT /managedClusters` et attribuez une classe de stockage par défaut différente à l' `DefaultStorageClass` paramètre.
> - Utilisez l'interface utilisateur Web Astra Control pour attribuer une classe de stockage par défaut différente. Voir Modifiez la classe de stockage par défaut.

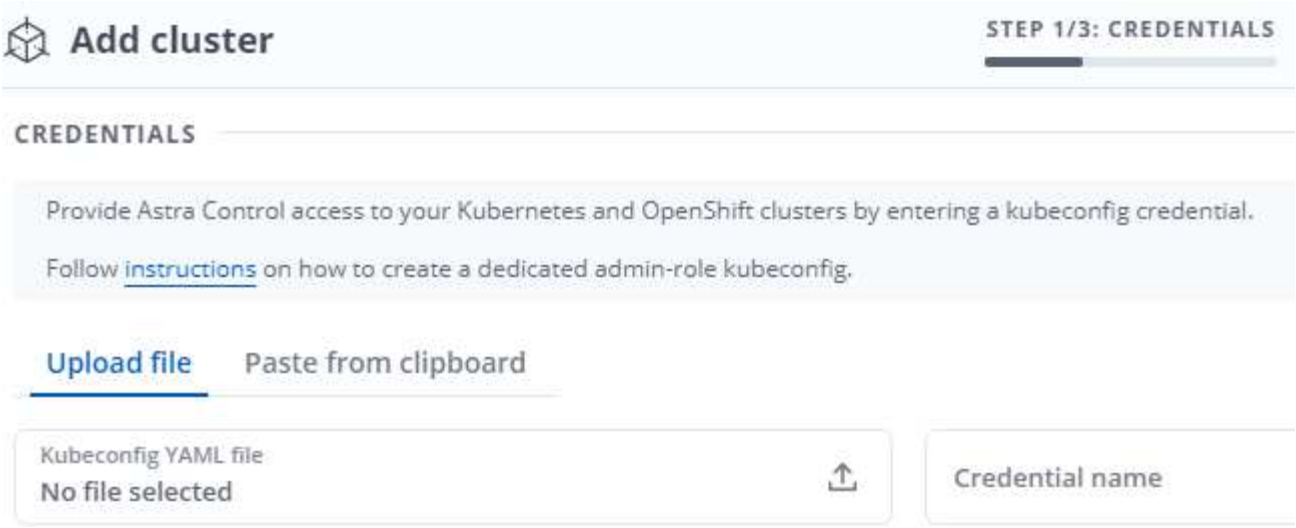
Ce dont vous avez besoin

- Avant d'ajouter un cluster, vérifiez et effectuez les opérations nécessaires "tâches préalables".

Étapes

1. Dans **Dashboard** de l'interface utilisateur du Centre de contrôle Astra, sélectionnez **Add** dans la section clusters.
2. Dans la fenêtre **Ajouter un cluster** qui s'ouvre, chargez un `kubeconfig.yaml` classez le contenu d'un `kubeconfig.yaml` fichier.

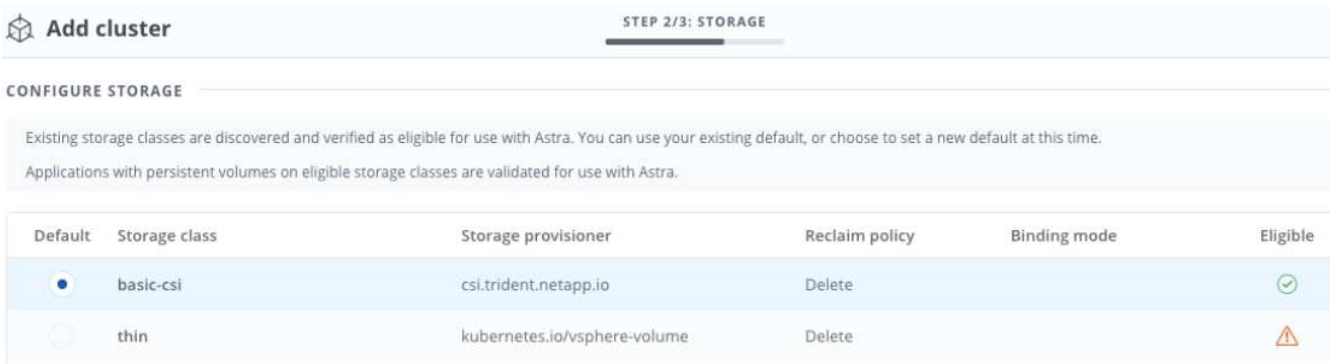
> Le `kubeconfig.yaml` le fichier doit inclure **uniquement les informations d'identification du cluster pour un cluster**.



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

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

> Nous vous recommandons de sélectionner une classe de stockage Trident avec le stockage ONTAP ou le magasin de données Astra.



6. Vérifiez les informations, et si tout semble bien, sélectionnez **Ajouter cluster**.

Résultat

Le cluster passe à l'état **découverte**, puis à **en cours d'exécution**. Vous avez ajouté un cluster Kubernetes et gérez-le dans 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.

Ajout d'un système back-end

Vous pouvez ajouter un système de stockage back-end pour qu'Astra Control puisse gérer ses ressources. Vous pouvez déployer un système back-end de stockage sur un cluster géré ou utiliser un système back-end existant.

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.

Il vous faudra pour déployer un data Store Astra

- Vous avez ajouté votre cluster d'applications Kubernetes et le cluster de calcul sous-jacent.

> Lorsque vous ajoutez votre cluster d'applications Kubernetes pour Astra Data Store et qu'il est géré par Astra Control, le cluster apparaît comme `unmanaged` dans la liste des systèmes back-end découverts. Vous devez ensuite ajouter le cluster de calcul qui contient Astra Data Store et qui intègre le cluster d'applications Kubernetes. Vous pouvez le faire à partir de **Backends** dans l'interface utilisateur. Sélectionnez le menu actions du cluster, puis `Manage`, et "ajouter le cluster". Après l'état du cluster de `unmanaged` Modifications au nom du cluster Kubernetes, vous pouvez procéder à l'ajout d'un back-end.

Il vous faudra de nouveaux déploiements de data Store Astra

- Vous avez "a chargé la version du pack d'installation que vous envisagez de déployer" À un endroit accessible à Astra Control.
- Vous avez ajouté le cluster Kubernetes que vous souhaitez utiliser pour le déploiement.
- Vous avez téléchargé le Licence Astra Data Store Pour votre déploiement vers un emplacement accessible à Astra Control.

Options

- Déploiement des ressources de stockage
- Utiliser un système back-end existant

Déploiement des ressources de stockage

Vous pouvez déployer un nouveau magasin de données Astra et gérer le stockage back-end associé.

Étapes

1. Naviguer dans le tableau de bord ou le menu Backends :
    - Dans **Dashboard** : dans le Résumé des ressources, sélectionnez un lien dans le volet stockage arrière-plans et sélectionnez **Ajouter** dans la section Backends.

- À partir de **Backends** :
  i. Dans la zone de navigation de gauche, sélectionnez **Backends**.
  ii. Sélectionnez **Ajouter**.

2. Sélectionnez l'option de déploiement **Astra Data Store** dans l'onglet **Deploy**.
3. Sélectionnez le package de magasin de données Astra à déployer :
   a. Entrez un nom pour l'application de magasin de données Astra.
   b. Choisissez la version d'Astra que vous voulez déployer.

> Si vous n'avez pas encore téléchargé la version que vous avez l'intention de déployer, vous pouvez utiliser l'option **Ajouter un paquet** ou quitter l'assistant et utiliser "gestion des packages" pour télécharger le pack d'installation.

4. Sélectionnez une licence Astra Data Store que vous avez déjà téléchargée ou utilisez l'option **Ajouter une licence** pour télécharger une licence à utiliser avec l'application.

> Les licences Astra Data Store avec autorisation complète sont associées à votre cluster Kubernetes, et ces clusters associés doivent apparaître automatiquement. S'il n'y a pas de cluster géré, vous pouvez sélectionner l'option **Ajouter un cluster** pour en ajouter un à Astra Control Management. Pour les licences Astra Data Store, si aucune association n'a été établie entre la licence et le cluster, vous pouvez définir cette association à la page suivante de l'assistant.

5. Si vous n'avez pas ajouté de cluster Kubernetes à la gestion Astra Control, vous devez le faire depuis la page **cluster Kubernetes**. Sélectionnez un cluster existant dans la liste ou sélectionnez **Ajouter le cluster sous-jacent** pour ajouter un cluster à la gestion Astra Control.
6. Sélectionnez une taille de modèle pour le cluster Kubernetes qui fournira les ressources pour le magasin de données Astra. Vous pouvez choisir l'une des options suivantes :
   - Si vous le souhaitez `Recommended Kubernetes worker node requirements`, sélectionnez un modèle de grande à petite en fonction de ce que votre licence autorise.
   - Si vous le souhaitez `Custom Kubernetes worker node requirements`, sélectionnez le nombre de cœurs et la mémoire totale que vous souhaitez pour chaque nœud de cluster. Vous pouvez également afficher le nombre de nœuds éligibles qui répondent à vos critères de sélection pour les cœurs et la mémoire.

> Lorsque vous choisissez un modèle, sélectionnez des nœuds de grande taille avec plus de mémoire et de cœurs pour des charges de travail plus importantes ou un nombre plus important de nœuds pour des charges de travail plus petites. Vous devez sélectionner un modèle en fonction de ce que votre licence autorise. Chaque option de modèle recommandée indique le nombre de nœuds éligibles qui répondent au modèle de modèle pour la mémoire, les cœurs et la capacité de chaque nœud.

7. Configurez les nœuds :
   a. Ajoutez une étiquette de nœud pour identifier le pool de nœuds de travail qui prend en charge ce cluster de magasin de données Astra.

> L'étiquette doit être ajoutée à chaque nœud du cluster qui sera utilisé pour le déploiement du magasin de données Astra avant le début du déploiement, sinon le déploiement échouera.

b. Configurez la capacité (Gio) par nœud manuellement ou sélectionnez la capacité maximale de nœud autorisée.

c. Configurez un nombre maximum de nœuds autorisés dans le cluster ou autorisez le nombre maximum de nœuds sur le cluster.

8. (Licences complètes de l'Astra Data Store uniquement) Entrez la clé de l'étiquette que vous souhaitez utiliser pour les domaines de protection.

> Créez au moins trois étiquettes uniques pour la clé pour chaque nœud. Par exemple, si votre clé est `astra.datastore.protection.domain`, vous pouvez créer les étiquettes suivantes : `astra.datastore.protection.domain=domain1` ,`astra.datastore.protection.domain=domain2`, et `astra.datastore.protection.domain=domain3`.

9. Configurez le réseau de gestion :

   a. Saisissez une adresse IP de gestion pour la gestion interne du magasin de données Astra qui se trouve sur le même sous-réseau que les adresses IP du nœud de travail.

   b. Choisissez d'utiliser la même carte réseau à la fois pour les réseaux de gestion et de données ou de les configurer séparément.

   c. Entrez le pool d'adresses IP du réseau de données, le masque de sous-réseau et la passerelle pour l'accès au stockage.

10. Vérifiez la configuration et sélectionnez **Deploy** pour commencer l'installation.

Résultat

Après une installation réussie, le système back-end apparaît dans `available` état dans la liste des systèmes back-end avec des informations de performance actives.

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

Utiliser un système back-end existant

Vous pouvez intégrer un système back-end de stockage ONTAP ou Astra dans la gestion du centre de contrôle d'Astra.

Étapes

1. Naviguer dans le tableau de bord ou le menu Backends :
   - Dans **Dashboard** : dans le Résumé des ressources, sélectionnez un lien dans le volet stockage arrière-plans et sélectionnez **Ajouter** dans la section Backends.
   - À partir de **Backends** :
     i. Dans la zone de navigation de gauche, sélectionnez **Backends**.
     ii. Sélectionnez **Manage** sur un back-end découvert à partir du cluster géré ou sélectionnez **Add** pour gérer un back-end existant supplémentaire.
2. Sélectionnez l'onglet **utiliser existant**.
3. Effectuez l'une des opérations suivantes en fonction de votre type de système back-end :
   - **Magasin de données Astra**:
     i. Sélectionnez **Astra Data Store**.

ii. Sélectionnez le cluster de calcul géré et sélectionnez **Suivant**.

iii. Confirmez les détails du back-end et sélectionnez **Ajouter le back-end de stockage**.

- **ONTAP** :

  i. Sélectionnez **ONTAP** et sélectionnez **Suivant**.

  ii. Saisissez l'adresse IP de gestion du cluster ONTAP et les identifiants d'administrateur.

  > 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, activez les méthodes d'accès `ontapi` et `http` Pour l'utilisateur sur les deux clusters ONTAP. Voir "Gérer les comptes d'utilisateurs" pour en savoir plus.

  iii. Sélectionnez **Revue**.

  iv. Confirmez les détails du back-end et sélectionnez **Ajouter le back-end de stockage**.

Résultat

Le back-end apparaît dans `available` état dans la liste avec des informations récapitulatives.

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

Ajouter un godet

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.

Lorsque vous ajoutez un godet, Astra Control marque un godet comme indicateur de compartiment par défaut. Le premier compartiment que vous créez devient le compartiment par défaut.

Il n'est pas nécessaire de cloner la configuration de vos applications et le stockage persistant vers le même cluster.

Utiliser l'un des types de godet suivants :

- NetApp ONTAP S3
- NetApp StorageGRID S3
- S3 générique

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

- Microsoft Azure

  > Bien qu'Astra Control Center prenne en charge Amazon S3 en tant que fournisseur de compartiments S3 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.

- Microsoft Azure

Pour plus d'informations sur l'ajout de compartiments à l'aide de l'API de contrôle Astra, reportez-vous à la section "Informations sur l'automatisation et les API d'Astra".

Étapes

1. Dans la zone de navigation de gauche, sélectionnez **godets**.
   a. Sélectionnez **Ajouter**.
   b. 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.

   c. Créer un nouveau nom de compartiment ou saisir 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 ultérieurement lors de la création d'une sauvegarde. Ce nom apparaît également lors de la configuration de la règle de protection.

   d. Entrez le nom ou l'adresse IP du terminal S3.
   e. Si vous souhaitez que ce compartiment soit utilisé comme compartiment par défaut pour toutes les sauvegardes, vérifiez le `Make this bucket the default bucket for this private cloud` option.

   > Cette option n'apparaît pas pour le premier compartiment que vous créez.

   f. Continuez en ajoutant informations d'identification.

Ajoutez des identifiants d'accès S3

Ajoutez les identifiants d'accès S3 à tout moment.

Étapes

1. Dans la boîte de dialogue compartiments, sélectionnez l'onglet **Ajouter** ou **utiliser existant**.
   a. Saisissez un nom pour l'identifiant qui le distingue des autres identifiants dans Astra Control.
   b. Saisissez l'ID d'accès et la clé secrète en collant le contenu dans le presse-papiers.

Modifiez la classe de stockage par défaut

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

Étapes

1. Dans l'interface utilisateur Web Astra Control Center, 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**.

Et la suite ?

Maintenant que vous vous êtes connecté et que vous avez ajouté des clusters à Astra Control Center, vous pouvez commencer à utiliser les fonctions de gestion des données applicatives d'Astra Control Center.

- "Gérer les utilisateurs"
- "Commencez à gérer les applications"
- "Protégez vos applications"
- "Clonage des applications"
- "Gérer les notifications"
- "Connectez-vous à Cloud Insights"
- "Ajouter un certificat TLS personnalisé"

Trouvez plus d'informations

- "Utilisez l'API de contrôle Astra"
- "Problèmes connus"

Conditions préalables à l'ajout d'un cluster

Assurez-vous que les conditions préalables sont remplies avant d'ajouter un cluster. Vous devez également effectuer les vérifications d'admissibilité pour vous assurer que votre grappe est prête à être ajoutée au Centre de contrôle Astra.

Ce dont vous avez besoin avant d'ajouter un cluster

Assurez-vous que votre cluster répond aux exigences décrites dans "Configuration requise en cluster des applications".

> Si vous prévoyez d'ajouter un deuxième cluster OpenShift 4.6, 4.7 ou 4.8 en tant que ressource de calcul gérée, assurez-vous que la fonctionnalité Snapshot de volume Astra Trident est activée. Découvrez l'Astra Trident officielle "instructions" Pour activer et tester des copies Snapshot de volume avec Astra Trident.

- Les classes de stockage Astra Trident sont configurées avec un "système back-end pris en charge" (requis pour tout type de cluster)
- Le superutilisateur et l'ID utilisateur définis sur le système ONTAP de sauvegarde pour sauvegarder et restaurer des applications avec le Centre de contrôle Astra. Exécutez la commande suivante dans la ligne de commande ONTAP :
  ```
  export-policy rule modify -vserver <storage virtual machine name> -policyname
  <policy name> -ruleindex 1 -superuser sysm --anon 65534
  ```
- Découvrez Astra Trident `volumesnapshotclass` objet défini par un administrateur. Découvrez Astra

Trident "instructions" Pour activer et tester des copies Snapshot de volume avec Astra Trident.

- Assurez-vous de n'avoir qu'une seule classe de stockage par défaut définie pour votre cluster Kubernetes.

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. Vérifiez la version de Trident.

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

Si Trident est présent, vous voyez des valeurs de sortie similaires à celles illustrées dans l'exemple suivant :

```
NAME      VERSION
trident   21.04.0
```

Si Trident n'existe pas, vous voyez des résultats similaires à ce qui suit :

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

> Si Trident n'est pas installé ou si la version installée n'est pas la dernière, vous devez installer la dernière version de Trident avant de continuer. Voir la "Documentation Trident" pour obtenir des instructions.

2. Vérifiez si les classes de stockage utilisent les pilotes Trident pris en charge. Le nom de provisionnement doit être `csi.trident.netapp.io`. Voir l'exemple suivant :

```
kubectl get sc
NAME                   PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)   csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                   kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

Créez un kubeconfig. Rôle admin

Avant d'effectuer les étapes suivantes, assurez-vous que vous disposez des éléments suivants sur votre machine :

- `kubectl` v1.19 ou version ultérieure installé

- Un kubeconfig actif avec des droits d'administrateur de cluster pour le contexte actif

Étapes

1. Créer un compte de service comme suit :

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

      Ajustez le nom et l'espace de noms selon vos besoins. Si des modifications sont apportées ici, vous devez appliquer les mêmes modifications dans les étapes suivantes.

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

+

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

   a. Appliquer le compte de service :

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

2. (Facultatif) si votre cluster utilise une politique de sécurité de pod restrictive qui ne permet pas la création de pod privilégié ou l'exécution des processus dans les conteneurs de pod en tant qu'utilisateur racine, créez une politique de sécurité de pod personnalisée pour le cluster qui permet à Astra Control de créer et de gérer des pods. Pour obtenir des instructions, reportez-vous à la section "Créez une stratégie de sécurité de pod personnalisée".

3. Accordez des autorisations d'administration du cluster comme suit :

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

      Ajustez les noms et espaces de noms modifiés lors de la création du compte de service, le cas échéant.

```
<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: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

a. Appliquer la liaison de rôle de cluster :

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

4. 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-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

Les indices pour chaque élément dans `secrets` la matrice commence par 0. Dans l'exemple ci-dessus, l'index de `astracontrol-service-account-dockercfg-vhz87` serait 0 et l'index pour `astracontrol-service-account-token-r59kr` serait 1. Dans votre résultat, notez l'index du nom du compte de service qui contient le mot "jeton".

5. 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
```

6. (**Facultatif**) Renommer le kubeconfig en un nom significatif pour votre grappe. Protéger les informations d'identification du cluster.

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

Et la suite ?

Maintenant que vous avez vérifié que les conditions préalables sont remplies, vous êtes prêt à "ajouter un cluster".

Trouvez plus d'informations

- "Documentation Trident"
- "Utilisez l'API de contrôle Astra"

Ajouter un certificat TLS personnalisé

Vous pouvez supprimer le certificat TLS auto-signé existant et le remplacer par un certificat TLS signé par une autorité de certification (AC).

Ce dont vous avez besoin

- Cluster Kubernetes avec Astra Control Center installé
- Accès administratif à un shell de commande sur le cluster à exécuter `kubectl` commandes
- Clé privée et fichiers de certificat de l'autorité de certification

Supprimez le certificat auto-signé

Supprimez le certificat TLS auto-signé existant.

1. Avec SSH, connectez-vous au cluster Kubernetes qui héberge Astra Control Center en tant qu'utilisateur administratif.
2. Recherchez le code secret TLS associé au certificat en cours à l'aide de la commande suivante, remplacement `<ACC-deployment-namespace>` Avec l'espace de noms de déploiement d'Astra Control Center :

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. Supprimez le certificat et le secret actuellement installés à l'aide des commandes suivantes :

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

Ajoutez un nouveau certificat

Ajoutez un nouveau certificat TLS signé par une autorité de certification.

1. Utilisez la commande suivante pour créer le nouveau secret TLS avec la clé privée et les fichiers de certificat de l'autorité de certification, en remplaçant les arguments entre parenthèses <> par les informations appropriées :

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. Utilisez la commande et l'exemple suivants pour modifier le fichier CRD (Custom Resource Definition) du cluster et modifier `spec.selfSigned` valeur à `spec.ca.secretName` Pour consulter le secret TLS créé précédemment :

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
....

#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. Utilisez la commande suivante et exemple de résultat pour vérifier que les modifications sont correctes et le cluster est prêt à valider les certificats, en remplaçant `<ACC-deployment-namespace>` Avec l'espace de noms de déploiement d'Astra Control Center :

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Créer le `certificate.yaml` fichier avec l'exemple suivant, en remplaçant les valeurs de paramètre fictif entre parenthèses <> par les informations appropriées :

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Créez le certificat à l'aide de la commande suivante :

```
kubectl apply -f certificate.yaml
```

6. À l'aide de la commande et de l'exemple de sortie suivants, vérifiez que le certificat a été créé correctement et avec les arguments que vous avez spécifiés lors de la création (tels que le nom, la durée, la date limite de renouvellement et les noms DNS).

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Modifiez l'option Ingress CRD TLS pour pointer vers votre nouveau secret de certificat à l'aide de la commande suivante et de l'exemple, en remplaçant les valeurs de paramètre fictif entre parenthèses <> par les informations appropriées :

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. À l'aide d'un navigateur Web, accédez à l'adresse IP de déploiement d'Astra Control Center.
9. Vérifiez que les détails du certificat correspondent aux détails du certificat que vous avez installé.
10. Exportez le certificat et importez le résultat dans le gestionnaire de certificats de votre navigateur Web.

Créez une stratégie de sécurité de pod personnalisée

Astra Control doit créer et gérer des pods Kubernetes sur les clusters qu'il gère. Si votre cluster utilise une politique de sécurité de pod restrictive qui ne permet pas la création de pod privilégié ou l'exécution des processus dans les conteneurs de pod en tant qu'utilisateur racine, vous devez créer une stratégie de sécurité de pod moins restrictive pour permettre à Astra Control de créer et de gérer ces pods.

Étapes

1. Créez une politique de sécurité du pod pour le cluster qui est moins restrictive par défaut et enregistrez-la dans un fichier. Par exemple :

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. Créez un nouveau rôle pour la stratégie de sécurité du pod.

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. Lier le nouveau rôle au compte de service.

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```