Poser le centre de contrôle Astra

Astra Control Center

NetApp
August 11, 2025

Sommaire

Installer le centre de contrôle Astra en suivant la procédure standard

Pour installer Astra Control Center, téléchargez les images d'installation et effectuez les étapes suivantes. Vous pouvez utiliser cette procédure pour installer Astra Control Center dans des environnements connectés à Internet ou équipés d'un filtre à air.

Pour une démonstration du processus d'installation d'Astra Control Center, reportez-vous àla section "vidéo".

Avant de commencer

- **Respecter les conditions préalables environnementales** : "Avant de commencer l'installation, préparez votre environnement pour le déploiement d'Astra Control Center".

  Déployez Astra Control Center dans un troisième domaine de panne ou sur un site secondaire. Cela est recommandé pour la réplication d'applications et la reprise sur incident transparente.

- **Assurer des services sains** : vérifier que tous les services API sont en bon état et disponibles :

```
kubectl get apiservices
```

- **Assurez-vous qu'un FQDN routable** : le FQDN Astra que vous prévoyez d'utiliser peut être routé vers le cluster. Cela signifie que vous avez une entrée DNS dans votre serveur DNS interne ou que vous utilisez une route URL de base déjà enregistrée.
- **Configurer cert Manager** : si un gestionnaire de certificats existe déjà dans le cluster, vous devez en effectuer quelques-uns "étapes préalables" Pour qu'Astra Control Center ne tente pas d'installer son propre gestionnaire de certificat. Par défaut, Astra Control Center installe son propre gestionnaire de certificats lors de l'installation.
- * (Pilote SAN ONTAP uniquement) Activer le multipath* : si vous utilisez un pilote SAN ONTAP, assurez-vous que le multipath est activé sur tous vos clusters Kubernetes.

Vous devez également tenir compte des points suivants :

- **Accéder au registre d'images NetApp Astra Control** :

  Vous avez la possibilité d'obtenir des images d'installation et des améliorations de fonctionnalités pour Astra Control, telles que Astra Control Provisioner, à partir du registre d'images NetApp.

  a. Notez l'ID de votre compte Astra Control dont vous aurez besoin pour vous connecter au registre.

     Votre ID de compte s'affiche dans l'interface utilisateur web d'Astra Control Service. Sélectionnez l'icône de figure en haut à droite de la page, sélectionnez **API Access** et notez votre ID de compte.

  b. 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.

  c. Connectez-vous au registre Astra Control :

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

- **Installer un maillage de services pour des communications sécurisées** : il est fortement recommandé de sécuriser les canaux de communication du cluster hôte Astra Control à l'aide d'un "maillage de service pris en charge".

  L'intégration d'Astra Control Center avec un maillage de service ne peut être effectuée que dans Astra Control Center "installation" et pas indépendant de ce processus. Le retour d'un environnement maillé à un environnement non maillé n'est pas pris en charge.

  Pour l'utilisation du maillage de service Istio, vous devez effectuer les opérations suivantes :

  - Ajouter un `istio-injection:enabled` étiquette Dans l'espace de noms Astra avant de déployer Astra Control Center.
  - Utilisez le `Generic` paramètre d'entrée et fournissent une entrée alternative pour équilibrage de la charge externe.
  - Pour les clusters Red Hat OpenShift, vous devez définir `NetworkAttachmentDefinition` Sur tous les espaces de noms Astra Control Center associés (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` pour les clusters d'applications, ou tout espace de noms personnalisé ayant été substitué).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Étapes

Pour installer le centre de contrôle Astra, procédez comme suit :

- Téléchargez et extrayez Astra Control Center
- Suivez les étapes supplémentaires si vous utilisez un registre local
- Configurez l'espace de noms et le secret pour les registres avec les exigences d'authentification
- Poser le conducteur du centre de commande Astra
- Configurer le centre de contrôle Astra
- Installation complète du centre de contrôle Astra et du conducteur
- Vérifiez l'état du système
- Configurer l'entrée pour l'équilibrage de charge
- Connectez-vous à l'interface utilisateur du centre de contrôle Astra

> Ne supprimez pas l'opérateur du centre de contrôle Astra (par exemple, `kubectl delete -f astra_control_center_operator_deploy.yaml`) À tout moment pendant l'installation ou le fonctionnement d'Astra Control Center pour éviter de supprimer les modules.

Téléchargez et extrayez Astra Control Center

Téléchargez les images d'Astra Control Center à partir de l'un des emplacements suivants :

- **Registre d'images du service Astra Control** : utilisez cette option si vous n'utilisez pas de registre local avec les images d'Astra Control Center ou si vous préférez cette méthode au téléchargement du bundle à partir du site de support NetApp.
- **Site de support NetApp** : utilisez cette option si vous utilisez un registre local avec les images du Centre de contrôle Astra.

Registre d'images Astra Control

1. Connectez-vous à Astra Control Service.
2. Sur le tableau de bord, sélectionnez **Deploy a autogéré instance d'Astra Control**.
3. Suivez les instructions pour vous connecter au registre d'images Astra Control, extraire l'image d'installation d'Astra Control Center et extraire l'image.

Site de support NetApp

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

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

```
openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
-signature certs/astra-control-center-[version].tar.gz.sig astra-
control-center-[version].tar.gz
```

   La sortie s'affiche `Verified OK` une fois la vérification terminée.

3. Extraire les images du pack Astra Control Center :

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

Suivez les étapes supplémentaires si vous utilisez un registre local

Si vous prévoyez d'envoyer le bundle Astra Control Center vers votre registre local, vous devez utiliser le plug-in de ligne de commande NetApp Astra kubectl.

Installez le plug-in NetApp Astra kubectl

Procédez comme suit pour installer le dernier plug-in de ligne de commande NetApp Astra kubectl.

Avant de commencer

NetApp fournit des binaires de plug-ins pour différentes architectures CPU et systèmes d'exploitation. Avant d'effectuer cette tâche, vous devez savoir quelle unité centrale et quel système d'exploitation vous possédez.

Si vous avez déjà installé le plug-in à partir d'une installation précédente, "vérifiez que vous disposez de la dernière version" avant d'effectuer ces étapes.

Étapes

1. Répertoriez les binaires kubectl du plug-in NetApp Astra disponibles :

   La bibliothèque de plug-ins kubectl fait partie du bundle tar et est extraite dans le dossier `kubectl-astra`.

   ```
   ls kubectl-astra/
   ```

2. Déplacez le fichier dont vous avez besoin pour votre système d'exploitation et votre architecture CPU dans le chemin actuel et renommez-le `kubectl-astra`:

   ```
   cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
   ```

Ajoutez les images à votre registre

1. Si vous prévoyez d'envoyer le bundle Astra Control Center vers votre registre local, suivez la séquence d'étapes appropriée pour votre moteur de mise en conteneurs :

Docker

a. Accédez au répertoire racine du tarball. Vous devriez voir le `acc.manifest.bundle.yaml` et les répertoires suivants :

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Envoyez les images du package dans le répertoire d'images Astra Control Center vers votre registre local. Effectuez les remplacements suivants avant d'exécuter le `push-images` commande :

- Remplacez <BUNDLE_FILE> par le nom du fichier bundle Astra Control (`acc.manifest.bundle.yaml`).
- Remplacer <MY_FULL_REGISTRY_PATH> par l’URL du référentiel Docker, par exemple "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
- Remplacez <MY_REGISTRY_USER> par le nom d'utilisateur.
- Remplacez <MY_REGISTRY_TOKEN> par un jeton autorisé pour le registre.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Accédez au répertoire racine du tarball. Vous devriez voir ce fichier et ce répertoire:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Connectez-vous à votre registre :

```
podman login <YOUR_REGISTRY>
```

c. Préparez et exécutez l'un des scripts suivants qui est personnalisé pour la version de Podman que vous utilisez. Remplacez <MY_FULL_REGISTRY_PATH> par l'URL de votre référentiel qui inclut tous les sous-répertoires.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

Le chemin d'accès à l'image que le script crée doit ressembler aux éléments suivants, selon la configuration de votre registre :

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Modifier le répertoire :

```
cd manifests
```

Configurez l'espace de noms et le secret pour les registres avec les exigences d'authentification

1. Exportez la configuration kubeconfig pour le cluster hôte Astra Control Center :

```
export KUBECONFIG=[file path]
```

> Avant de terminer l'installation, assurez-vous que votre kubeconfig pointe vers le cluster où vous souhaitez installer Astra Control Center.

2. Si vous utilisez un registre qui nécessite une authentification, vous devez procéder comme suit :

   a. Créer le `netapp-acc-operator` espace de noms :

```
kubectl create ns netapp-acc-operator
```

   b. Créez un secret pour le `netapp-acc-operator` espace de noms. Ajoutez des informations sur Docker et exécutez la commande suivante :

> Le paramètre fictif `your_registry_path` doit correspondre à l'emplacement des images que vous avez téléchargées précédemment (par exemple, `[Registry_URL]/netapp/astra/astracc/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

+

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

+

> Si vous supprimez l'espace de noms après la génération du secret, recréez l'espace de noms, puis régénérez le secret pour l'espace de noms.

   a. Créer le `netapp-acc` (ou espace de nom personnalisé).

```
kubectl create ns [netapp-acc or custom namespace]
```

b. Créez un secret pour le `netapp-acc` (ou espace de nom personnalisé). Ajoutez des informations relatives à Docker et exécutez l'une des commandes appropriées en fonction de vos préférences de registre :

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=cr.astra.netapp.io --docker-username=[astra_account_id] --docker-password=[astra_api_token]
```

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=[your_registry_path] --docker-username=[username] --docker-password=[token]
```

Poser le conducteur du centre de commande Astra

1. (Registres locaux uniquement) si vous utilisez un registre local, procédez comme suit :

   a. Ouvrez le déploiement de l'opérateur Astra Control Center YAML :

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Un échantillon annoté YAML suit ces étapes.

   b. Si vous utilisez un registre qui nécessite une authentification, remplacez la ligne par défaut de `imagePullSecrets: []` avec les éléments suivants :

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Changer `ASTRA_IMAGE_REGISTRY` pour le `kube-rbac-proxy` image dans le chemin du registre où vous avez poussé les images dans un étape précédente.

   d. Changer `ASTRA_IMAGE_REGISTRY` pour le `acc-operator-controller-manager` image dans le chemin du registre où vous avez poussé les images dans un étape précédente.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
```

```
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
```

```
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

2. Poser le conducteur du centre de commande Astra :

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

Développer pour une réponse d'échantillon :

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.as
tra.netapp.io created
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
created
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role
created
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding created
configmap/acc-operator-manager-config created
service/acc-operator-controller-manager-metrics-service created
deployment.apps/acc-operator-controller-manager created
```

3. Vérifiez que les pods sont en cours d'exécution :

```
kubectl get pods -n netapp-acc-operator
```

Configurer le centre de contrôle Astra

1. Modifiez le fichier de ressources personnalisées (CR) Astra Control Center (`astra_control_center.yaml`) pour créer des comptes, un support, un registre et d'autres configurations nécessaires :

```
vim astra_control_center.yaml
```

Un échantillon annoté YAML suit ces étapes.

2. Modifiez ou confirmez les paramètres suivants :

Nom du compte

Réglage	Guidage	Type	Exemple
`accountName`	Modifiez le `accountName` Chaîne du nom que vous souhaitez associer au compte Astra Control Center. Il ne peut y avoir qu'un seul nom de compte.	chaîne	`Example`

AstraVersion

Réglage	Guidage	Type	Exemple
`astraVersion`	La version d'Astra Control Center à déployer. Aucune action n'est nécessaire pour ce paramètre car la valeur sera pré-remplie.	chaîne	`24.02.0-69`

Adresse postale

Réglage	Guidage	Type	Exemple
`astraAddress`	Modifiez le `astraAddress` Chaîne sur le FQDN (recommandé) ou l'adresse IP que vous souhaitez utiliser dans votre navigateur pour accéder à Astra Control Center. Cette adresse définit la façon dont Astra Control Center se trouve dans votre centre de données et est le même FQDN ou l'adresse IP que vous avez fournie à partir de votre équilibreur de charge une fois que vous avez terminé "Exigences du centre de contrôle Astra". REMARQUE : ne pas utiliser `http://` ou `https://` dans l'adresse. Copier ce FQDN pour l'utiliser dans un plus tard.	chaîne	`astra.example.com`

AutoSupport

Vos sélections dans cette section déterminent si vous participerez à l'application de support proactif de NetApp, au conseiller numérique et à l'emplacement d'envoi des données. Une connexion Internet est requise (port 442) et toutes les données de support sont anonymisées.

Réglage	Utiliser	Guidage	Type	Exemple
`autoSupport.enrolled`	Soit `enrolled` ou `url` les champs doivent être sélectionnés	Changer `enrolled` Pour AutoSupport à `false` pour les sites sans connexion internet ou sans conservation `true` pour les sites connectés. Un réglage de `true` Les données anonymes peuvent être envoyées à NetApp pour bénéficier d'un support. La sélection par défaut est `false` Aucune donnée de support n'est envoyée à NetApp.	Booléen	`false` (cette valeur est la valeur par défaut)
`autoSupport.url`	Soit `enrolled` ou `url` les champs doivent être sélectionnés	Cette URL détermine l'emplacement d'envoi des données anonymes.	chaîne	`https://support.netapp.com/asupprod/post/1.0/postAsup`

e-mail

Réglage	Guidage	Type	Exemple
`email`	Modifiez le `email` chaîne à l'adresse d'administrateur initiale par défaut. Copiez cette adresse e-mail pour l'utiliser dans un plus tard. Cette adresse e-mail sera utilisée comme nom d'utilisateur du compte initial pour se connecter à l'interface utilisateur et sera informée des événements dans Astra Control.	chaîne	`admin@example.com`

Prénom

Réglage	Guidage	Type	Exemple
`firstName`	Prénom de l'administrateur initial par défaut associé au compte Astra. Le nom utilisé ici sera visible dans un en-tête de l'interface utilisateur après votre première connexion.	chaîne	`SRE`

Nom de famille

Réglage	Guidage	Type	Exemple
`lastName`	Nom de l'administrateur initial par défaut associé au compte Astra. Le nom utilisé ici sera visible dans un en-tête de l'interface utilisateur après votre première connexion.	chaîne	`Admin`

Registre d'imageRegistry

Vos sélections dans cette section définissent le registre d'images du conteneur qui héberge les images d'application Astra, l'opérateur du centre de contrôle Astra et le référentiel Helm d'Astra Control Center.

Réglage	Utiliser	Guidage	Type	Exemple
`imageRegistry.name`	Obligatoire	Nom du registre d'images Astra Control qui héberge toutes les images requises pour déployer Astra Control Center. La valeur sera pré-remplie et aucune action n'est requise sauf si vous avez configuré un registre local. Dans le cas d'un registre local, remplacez cette valeur existante par le nom du registre d'images où vous avez poussé les images dans le étape précédente. Ne pas utiliser `http://` ou `https://` dans le nom du registre.	chaîne	`cr.astra.netapp.io` (valeur par défaut) `example.registry.com/astra` (exemple de registre local)

Réglage	Utiliser	Guidage	Type	Exemple
`imageRegistry.secret`	Facultatif	Nom du secret Kubernetes utilisé pour s'authentifier auprès du registre d'images. La valeur sera pré-remplie et aucune action n'est requise sauf si vous avez configuré un registre local et la chaîne que vous avez saisie pour ce registre dans `imageRegistry.name` requiert un secret. IMPORTANT : si vous utilisez un registre local qui ne nécessite pas d'autorisation, vous devez le supprimer `secret` ligne comprise entre `imageRegistry` sinon, l'installation échouera.	chaîne	`astra-registry-cred`

Classe de stockage

Réglage	Guidage	Type	Exemple
`storageClass`	Modifiez le `storageClass` valeur à partir de `ontap-gold` À une autre ressource de classe de stockage, comme requis par votre installation. Lancer la commande `kubectl get sc` pour déterminer vos classes de stockage configurées existantes. L'une des classes de stockage configurées par Astra Control Provisioner doit être saisie dans le fichier manifeste (`astra-control-center-<version>.manifest`) Et sera utilisé pour ASTRA PVS. Si elle n'est pas définie, la classe de stockage par défaut sera utilisée. REMARQUE : si une classe de stockage par défaut est configurée, assurez-vous qu'elle est la seule classe de stockage à avoir l'annotation par défaut.	chaîne	`ontap-gold`

Volume ReclaimPolicy

Réglage	Guidage	Type	Options
`volumeReclaimPolicy`	Cette règle définit la règle de récupération pour les volumes persistants d'Astra. Définition de cette règle sur `Retain` Conserve les volumes persistants après la suppression d'Astra. Définition de cette règle sur `Delete` supprime les volumes persistants après la suppression d'astra. Si cette valeur n'est pas définie, les PV sont conservés.	chaîne	• `Retain` (Il s'agit de la valeur par défaut) • `Delete`

Type d’esseType

Réglage	Guidage	Type	Options
`ingressType`	Utilisez l'un des types d'entrées suivants : **Générique** (`ingressType: "Generic"`) (Par défaut) Utilisez cette option si vous avez un autre contrôleur d'entrée en service ou si vous préférez utiliser votre propre contrôleur d'entrée. Une fois Astra Control Center déployée, vous devez configurer le "contrôleur d'entrée" Pour exposer Astra Control Center avec une URL. IMPORTANT : si vous avez l'intention d'utiliser un maillage de service avec Astra Control Center, vous devez sélectionner `Generic` comme type d'entrée et configurez votre propre "contrôleur d'entrée". **AccTraefik** (`ingressType: "AccTraefik"`) Utilisez cette option lorsque vous préférez ne pas configurer de contrôleur d'entrée. Ceci déploie le centre de contrôle Astra `traefik` Passerelle en tant que service de type Kubernetes LoadBalancer. Le centre de contrôle Astra utilise un service de type « équilibreur de charge » (`svc/traefik` Dans l'espace de noms du centre de contrôle Astra), et exige qu'il se	chaîne	• `Generic` (il s'agit de la valeur par défaut) • `AccTraefik`

Taille de l'échelle

Réglage	Guidage	Type	Options
`scaleSize`	Par défaut, Astra utilisera la haute disponibilité (HA) `scaleSize` de `Medium`, Qui déploie la plupart des services en haute disponibilité et déploie plusieurs répliques pour assurer la redondance. Avec `scaleSize` comme `Small`, Astra réduira le nombre de répliques pour tous les services, à l'exception des services essentiels, afin de réduire la consommation. CONSEIL : `Medium` les déploiements se composent d'environ 100 pods (à l'exclusion des workloads transitoires). 100 modules sont basés sur une configuration à trois nœuds maîtres et trois nœuds workers). Tenez compte des contraintes de limite réseau par pod qui peuvent représenter un problème dans votre environnement, en particulier lors de l'examen des scénarios de reprise d'activité.	chaîne	• `Small` • `Medium` (Il s'agit de la valeur par défaut)

AstressourcesScaler

Réglage	Guidage	Type	Options
`astraResourcesScaler`	Options d'évolutivité pour les limites de ressources AstrakControlCenter. Par défaut, Astra Control Center se déploie avec des demandes de ressources définies pour la plupart des composants d'Astra. Avec cette configuration, la pile logicielle Astra Control Center est plus performante dans les environnements soumis à une charge et à une évolutivité accrues des applications. Cependant, dans les scénarios utilisant des grappes de développement ou de test plus petites, le champ CR `astraResourcesScalar` peut être réglé sur `Off`. Cela désactive les demandes de ressources et permet un déploiement sur les clusters plus petits.	chaîne	• `Default` (Il s'agit de la valeur par défaut) • `Off`

Autres vals

Ajoutez les valeurs supplémentaires suivantes à l'Astra Control Center CR pour éviter un problème connu lors de l'installation :

```
additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

crds

Vos sélections dans cette section déterminent comment Astra Control Center doit traiter les CRD.

Réglage	Guidage	Type	Exemple
`crds.externalCertManager`	Si vous utilisez un gestionnaire de certificats externe, modifiez-le `externalCertManager` à `true`. La valeur par défaut `false` Provoque l'installation d'Astra Control Center de ses propres CRD de cert Manager lors de l'installation. Les CRDS sont des objets à l'échelle du cluster et leur installation peut avoir un impact sur d'autres parties du cluster. Vous pouvez utiliser cet indicateur pour signaler à Astra Control Center que ces CRD seront installés et gérés par l'administrateur de cluster en dehors du centre de contrôle Astra.	Booléen	`False` (cette valeur est la valeur par défaut)
`crds.externalTraefik`	Par défaut, Astra Control Center installe les CRD Traefik requis. Les CRDS sont des objets à l'échelle du cluster et leur installation peut avoir un impact sur d'autres parties du cluster. Vous pouvez utiliser cet indicateur pour signaler à Astra Control Center que ces CRD seront installés et gérés par l'administrateur de cluster en dehors du centre de contrôle Astra.	Booléen	`False` (cette valeur est la valeur par défaut)

Assurez-vous d'avoir sélectionné la classe de stockage et le type d'entrée appropriés pour votre configuration avant de terminer l'installation.

exemple astra_control_center.yaml

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[cr.astra.netapp.io or your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

Installation complète du centre de contrôle Astra et du conducteur

1. Si vous ne l'avez pas déjà fait dans une étape précédente, créez le `netapp-acc` (ou personnalisée) espace de noms :

```
kubectl create ns [netapp-acc or custom namespace]
```

2. Si vous utilisez un maillage de service avec Astra Control Center, ajoutez l'étiquette suivante au `netapp-acc` ou un espace de noms personnalisé :

   Votre type d'entrée (`ingressType`) doit être défini sur `Generic` Dans Astra Control Center CR avant de passer à cette commande.

   ```
   kubectl label ns [netapp-acc or custom namespace] istio-
   injection:enabled
   ```

3. (Recommandé) "Activez les licences MTL strictes" Pour le maillage de service Istio :

   ```
   kubectl apply -n istio-system -f - <<EOF
   apiVersion: security.istio.io/v1beta1
   kind: PeerAuthentication
   metadata:
     name: default
   spec:
     mtls:
       mode: STRICT
   EOF
   ```

4. Poser le centre de contrôle Astra dans le `netapp-acc` (ou votre espace de noms personnalisé) :

   ```
   kubectl apply -f astra_control_center.yaml -n [netapp-acc or custom
   namespace]
   ```

   L'opérateur d'Astra Control Center effectue une vérification automatique des exigences de l'environnement. Manquant "de formation" Peut entraîner une défaillance de votre installation ou un dysfonctionnement d'Astra Control Center. Voir la section suivante pour vérifier la présence de messages d'avertissement liés au contrôle automatique du système.

Vérifiez l'état du système

Vous pouvez vérifier l'état du système à l'aide des commandes kubectl. Si vous préférez utiliser OpenShift, vous pouvez utiliser des commandes oc comparables pour les étapes de vérification.

Étapes

1. Vérifiez que le processus d'installation n'a pas produit de messages d'avertissement relatifs aux vérifications de validation :

   ```
   kubectl get acc [astra or custom Astra Control Center CR name] -n
   [netapp-acc or custom namespace] -o yaml
   ```

Des messages d'avertissement supplémentaires sont également signalés dans les journaux de l'opérateur d'Astra Control Center.

2. Corrigez tous les problèmes de votre environnement qui ont été signalés par les vérifications automatisées des exigences.

Vous pouvez corriger les problèmes en vous assurant que votre environnement respecte les "de formation" Pour Astra Control Center.

3. Vérifiez que tous les composants du système sont correctement installés.

```
kubectl get pods -n [netapp-acc or custom namespace]
```

Chaque pod doit avoir un statut de `Running`. Le déploiement des modules du système peut prendre plusieurs minutes.

Développez pour obtenir une réponse d'échantillon

```
acc-helm-repo-5bd77c9ddd-8wxm2              1/1     Running     0
1h
activity-5bb474dc67-8l9ss                   1/1     Running     0
1h
activity-5bb474dc67-qbrtq                   1/1     Running     0
1h
api-token-authentication-6wbj2              1/1     Running     0
1h
api-token-authentication-9pgw6              1/1     Running     0
1h
api-token-authentication-tqf6d              1/1     Running     0
1h
asup-5495f44dbd-z4kft                       1/1     Running     0
1h
authentication-6fdd899858-5x45s             1/1     Running     0
1h
bucketservice-84d47487d-n9xgp               1/1     Running     0
1h
bucketservice-84d47487d-t5jhm               1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc               1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf               1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf    1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck    1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x        1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn        1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj                1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx                1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw    0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv            1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km            1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92       1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f          1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2          1/1     Running     0
1h
composite-volume-bff6f4f76-789nj            1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4            1/1     Running     0
1h
credentials-79fd64f788-m7m8f                1/1     Running     0
1h
credentials-79fd64f788-qnc6c                1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                 1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                 1/1     Running     0
1h
features-7b9585444c-7xd7m                   1/1     Running     0
1h
features-7b9585444c-dcqwc                   1/1     Running     0
1h
fluent-bit-ds-crq8m                         1/1     Running     0
1h
fluent-bit-ds-gmgq8                         1/1     Running     0
1h
fluent-bit-ds-gzr4f                         1/1     Running     0
1h
fluent-bit-ds-j6sf6                         1/1     Running     0
1h
fluent-bit-ds-v4t9f                         1/1     Running     0
1h
fluent-bit-ds-x7j59                         1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr             1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd             1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                 1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                 1/1     Running     0
1h
identity-95df98cb5-dvlmz                    1/1     Running     0
1h
identity-95df98cb5-krf59                    1/1     Running     0
1h
influxdb2-0                                 1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b          1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                    1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                    1/1     Running     0
1h
license-689cdd4595-2gsc8                    1/1     Running     0
1h
license-689cdd4595-g6vwk                    1/1     Running     0
1h
login-ui-57bb599956-4fwgz                   1/1     Running     0
1h
login-ui-57bb599956-rhztb                   1/1     Running     0
1h
loki-0                                      1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm             1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl             1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl        2/2     Running     0
1h
nats-0                                      1/1     Running     0
1h
nats-1                                      1/1     Running     0
1h
nats-2                                      1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd            1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                   1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                   1/1     Running     0
1h
openapi-864584bccc-75nlv                    1/1     Running     0
1h
openapi-864584bccc-zh6bx                    1/1     Running     0
1h
polaris-consul-consul-server-0              1/1     Running     0
1h
polaris-consul-consul-server-1              1/1     Running     0
1h
polaris-consul-consul-server-2              1/1     Running     0
1h
polaris-keycloak-0                          1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-815zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

4. (En option) regarder le `acc-operator` journaux de suivi de la progression :

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> `accHost` l'enregistrement du cluster est l'une des dernières opérations. en cas de défaillance, le déploiement ne pourra pas échouer. Dans l'éventualité où un échec d'enregistrement du cluster était indiqué dans les journaux, vous pouvez essayer de nouveau l'enregistrement via le "Ajout du flux de travail du cluster dans l'interface utilisateur" Ou API.

5. Lorsque tous les modules sont en cours d'exécution, vérifiez que l'installation a réussi (`READY` est `True`) Et obtenez le mot de passe de configuration initial que vous utiliserez lorsque vous vous connecterez à Astra Control Center :

```
kubectl get AstraControlCenter -n [netapp-acc or custom namespace]
```

Réponse :

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  24.02.0-69
10.111.111.111  True
```

Copiez la valeur UUID. Le mot de passe est `ACC-` Suivi de la valeur UUID (`ACC-[UUID]` ou, dans cet exemple, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Configurer l'entrée pour l'équilibrage de charge

Vous pouvez configurer un contrôleur d'entrée Kubernetes qui gère l'accès externe aux services. Ces procédures fournissent des exemples de configuration pour un contrôleur d'entrée si vous avez utilisé la valeur par défaut de `ingressType: "Generic"` Dans la ressource personnalisée Astra Control Center (`astra_control_center.yaml`). Vous n'avez pas besoin d'utiliser cette procédure si vous avez spécifié `ingressType: "AccTraefik"` Dans la ressource personnalisée Astra Control Center (`astra_control_center.yaml`).

Une fois Astra Control Center déployé, vous devez configurer le contrôleur Ingress pour exposer Astra Control Center avec une URL.

Les étapes de configuration varient en fonction du type de contrôleur d'entrée utilisé. Le centre de contrôle Astra prend en charge de nombreux types de contrôleurs d'entrée. Ces procédures de configuration fournissent des exemples d'étapes pour certains types de contrôleurs d'entrée courants.

Avant de commencer

- Le requis "contrôleur d'entrée" doit déjà être déployé.
- Le "classe d'entrée" correspondant au contrôleur d'entrée doit déjà être créé.

Étapes pour l'entrée Istio

1. Configurer l'entrée Istio.

   Cette procédure suppose que Istio est déployé à l'aide du profil de configuration par défaut.

2. Rassemblez ou créez le certificat et le fichier de clé privée souhaités pour la passerelle d'entrée.

   Vous pouvez utiliser un certificat signé par une autorité de certification ou auto-signé. Le nom commun doit être l'adresse Astra (FQDN).

   Exemple de commande :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

3. Créez un secret `tls secret name` de type `kubernetes.io/tls` Pour une clé privée TLS et un certificat dans `istio-system namespace` Comme décrit dans les secrets TLS.

   Exemple de commande :

```
kubectl create secret tls [tls secret name] --key="tls.key"
--cert="tls.crt" -n istio-system
```

   Le nom du secret doit correspondre au `spec.tls.secretName` fourni dans `istio-ingress.yaml` fichier.

4. Déployer une ressource d'entrée dans le `netapp-acc` (ou nom personnalisé) de l'espace de noms utilisant le type de ressource v1 pour un schéma (`istio-Ingress.yaml` est utilisé dans cet exemple) :

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Appliquer les modifications :

```
kubectl apply -f istio-Ingress.yaml
```

6. Vérifier l'état de l'entrée :

```
kubectl get ingress -n [netapp-acc or custom namespace]
```

Réponse :

```
NAME    CLASS HOSTS             ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

7. Terminer l'installation du centre de contrôle Astra.

Étapes du contrôleur d'entrée Nginx

1. Créer un secret de type `kubernetes.io/tls` Pour une clé privée TLS et un certificat dans `netapp-acc` (ou espace de noms personnalisé) comme décrit dans "Secrets TLS".
2. Déployez une ressource entrée dans `netapp-acc` (ou nom personnalisé) de l'espace de noms utilisant le type de ressource v1 pour un schéma (`nginx-Ingress.yaml` est utilisé dans cet exemple) :

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

3. Appliquer les modifications :

```
kubectl apply -f nginx-Ingress.yaml
```

NetApp recommande d'installer le contrôleur nginx en tant que déploiement plutôt qu'en tant que `daemonSet`.

Étapes du contrôleur d'entrée OpenShift

1. Procurez-vous votre certificat et obtenez les fichiers de clé, de certificat et d'autorité de certification prêts à l'emploi par la route OpenShift.
2. Création de la route OpenShift :

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Connectez-vous à l'interface utilisateur du centre de contrôle Astra

Après avoir installé Astra Control Center, vous allez modifier le mot de passe de l'administrateur par défaut et vous connecter au tableau de bord de l'interface utilisateur d'Astra Control Center.

Étapes

1. Dans un navigateur, saisissez le nom de domaine complet (y compris le `https://` prefix) que vous avez utilisé dans `astraAddress` dans le `astra_control_center.yaml` CR quand Vous avez installé Astra Control Center.
2. Acceptez les certificats auto-signés si vous y êtes invité.

   Vous pouvez créer un certificat personnalisé après la connexion.

3. Dans la page de connexion à Astra Control Center, entrez la valeur que vous avez utilisée `email` dans `astra_control_center.yaml` CR quand Vous avez installé Astra Control Center, suivi du mot de passe de configuration initiale (`ACC-[UUID]`).

   Si vous saisissez trois fois un mot de passe incorrect, le compte admin est verrouillé pendant 15 minutes.

4. Sélectionnez **connexion**.
5. Modifiez le mot de passe lorsque vous y êtes invité.

   S'il s'agit de votre première connexion et que vous oubliez le mot de passe et qu'aucun autre compte d'utilisateur administratif n'a encore été créé, contactez "Support NetApp" pour obtenir de l'aide sur la récupération des mots de

6. (Facultatif) supprimez le certificat TLS auto-signé existant et remplacez-le par un "Certificat TLS personnalisé signé par une autorité de certification".

Dépanner l'installation

Si l'un des services est dans `Error` état, vous pouvez inspecter les journaux. Rechercher les codes de réponse API dans la plage 400 à 500. Ceux-ci indiquent l'endroit où un échec s'est produit.

Options

- Pour inspecter les journaux de l'opérateur de l'Astra Control Center, entrez ce qui suit :

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

- Pour vérifier la sortie de l'Astra Control Center CR :

```
kubectl get acc -n [netapp-acc or custom namespace] -o yaml
```

Autres procédures d'installation

- **Installer avec Red Hat OpenShift OperatorHub** : utilisez cette option "autre procédure" Pour installer Astra Control Center sur OpenShift à l'aide d'OperatorHub.
- **Installer dans le Cloud public avec Cloud Volumes ONTAP backend**: Utiliser "ces procédures" Pour installer Astra Control Center dans Amazon Web Services (AWS), Google Cloud Platform (GCP) ou Microsoft Azure avec un système de stockage principal Cloud Volumes ONTAP.

Et la suite

- (Facultatif) en fonction de votre environnement, effectuez l'installation complète après l'installation "étapes de configuration".
- "Une fois Astra Control Center installé, connectez-vous à l'interface utilisateur et modifiez votre mot de passe, vous pouvez configurer une licence, ajouter des clusters, activer l'authentification, gérer le stockage et ajouter des compartiments".

Configurez un gestionnaire de certificats externe

Si un gestionnaire de certificats existe déjà dans votre cluster Kubernetes, vous devez effectuer certaines étapes préalables afin qu'Astra Control Center n'installe pas son propre gestionnaire de certificats.

Étapes

1. Vérifiez qu'un gestionnaire de certificats est installé :

```
kubectl get pods -A | grep 'cert-manager'
```

Exemple de réponse :

```
cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
Running    0     6d5h
cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
Running    0     6d5h
cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
Running    0     6d5h
```

2. Créez une paire de certificats/clés pour le `astraAddress` FQDN :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

Exemple de réponse :

```
Generating a 2048 bit RSA private key
...................+++
.......................................+++
writing new private key to 'tls.key'
```

3. Créez un secret avec des fichiers générés précédemment :

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

Exemple de réponse :

```
secret/selfsigned-tls created
```

4. Créer un `ClusterIssuer` fichier qui est **exactement** le suivant mais qui comprend l'emplacement de l'espace de noms où votre `cert-manager` des pods sont installés :

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Exemple de réponse :

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Vérifiez que le `ClusterIssuer` s'est correctement installé. `Ready` doit être de `True` avant de pouvoir continuer :

```
kubectl get ClusterIssuer
```

Exemple de réponse :

```
NAME                        READY   AGE
astra-ca-clusterissuer      True    9s
```

6. Complétez le "Procédure d'installation d'Astra Control Center". Il y a un "Étape de configuration requise pour le groupe de centre de contrôle Astra YAML" Dans lequel vous modifiez la valeur CRD pour indiquer que le gestionnaire de certificats est installé en externe. Vous devez effectuer cette étape pendant l'installation pour que le centre de contrôle Astra reconnaisse le responsable du certificat externe.

Informations sur le copyright

Informations sur les marques commerciales