Poser le centre de contrôle Astra

Astra Control Center

NetApp
November 21, 2023

Sommaire

Installer le centre de contrôle Astra en suivant la procédure standard

Pour installer Astra Control Center, téléchargez le bundle d'installation depuis le site de support NetApp et effectuez les opérations suivantes. Vous pouvez utiliser cette procédure pour installer Astra Control Center dans des environnements connectés à Internet ou équipés d'un filtre à air.

Autres procédures d'installation

- **Installer avec RedHat OpenShift OperatorHub**: Utilisez ceci "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.

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

Ce dont vous avez besoin

- "Avant de commencer l'installation, préparez votre environnement pour le déploiement d'Astra Control Center".
- Si vous avez configuré ou que vous souhaitez configurer des stratégies de sécurité de pod dans votre environnement, familiarisez-vous avec les stratégies de sécurité de pod et leur incidence sur l'installation d'Astra Control Center. Voir "Comprendre les restrictions de la stratégie de sécurité du pod".
- Assurez-vous que tous les services API sont en état de santé et disponibles :

```
kubectl get apiservices
```

- Assurez-vous que le FQDN Astra que vous prévoyez d'utiliser est routable vers ce 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.
- Si un cert Manager existe déjà dans le cluster, vous devez en effectuer certaines "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.

Description de la tâche

Le processus d'installation d'Astra Control Center vous aide à :

- Poser les composants Astra dans le `netapp-acc` (ou espace de nom personnalisé).
- Créez un compte d'administrateur propriétaire Astra Control par défaut.
- Définissez une adresse e-mail d'utilisateur administratif et un mot de passe de configuration initiale par défaut. Ce rôle de propriétaire est attribué à cet utilisateur pour la première connexion à l'interface utilisateur.
- Vérifiez que toutes les POD Astra Control Center sont en cours d'exécution.
- Installez 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.

Étapes

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

- Téléchargez et extrayez Astra Control Center
- Installez le plug-in NetApp Astra kubectl
- Ajoutez les images à votre 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

Téléchargez et extrayez Astra Control Center

1. Accédez au "Page de téléchargement de l'évaluation Astra Control Center" Sur le site de support NetApp.
2. Téléchargez le pack contenant Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

4. Extraire les images du pack Astra Control Center :

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

Installez le plug-in NetApp Astra kubectl

Le plug-in de ligne de commande NetApp Astra kubectl permet de gagner du temps lors de l'exécution des tâches courantes associées au déploiement et à la mise à niveau d'Astra Control Center.

Ce dont vous avez besoin

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.

Étapes

1. Répertoriez les binaires NetApp Astra kubectl disponibles et notez le nom du fichier dont vous avez besoin pour votre système d'exploitation et votre architecture de processeur :

   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 bon binaire 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 local

1. Suivez la séquence d'étapes appropriée pour votre moteur de mise en conteneurs :

Docker

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

   ```
   acc.manifest.bundle.yaml
   acc/
   ```

2. 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

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

   ```
   acc.manifest.bundle.yaml
   acc/
   ```

2. Connectez-vous à votre registre :

   ```
   podman login <YOUR_REGISTRY>
   ```

3. 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=22.11.0-82
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=22.11.0-82
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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version`

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

1. Exporter le KUBECONFIG pour le groupe hôte du centre de contrôle Astra :

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

Avant de terminer l'installation, assurez-vous que votre KUBECONFIG pointe vers le groupe d'instruments où vous souhaitez installer le centre de contrôle Astra. Le KUBECONFIG ne peut contenir qu'un seul contexte.

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
   ```

   Réponse :

   ```
   namespace/netapp-acc-operator created
   ```

   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/22.11.0-82`).

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

   Exemple de réponse :

   ```
   secret/astra-registry-cred created
   ```

   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.

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

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

   Exemple de réponse :

```
namespace/netapp-acc created
```

d. Créez un secret pour le `netapp-acc` (ou espace de nom personnalisé). Ajoutez des informations sur Docker et exécutez la commande suivante :

```
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]
```

Réponse

```
secret/astra-registry-cred created
```

Poser le conducteur du centre de commande Astra

1. Modifier le répertoire :

```
cd manifests
```

2. Modifiez le YAML de déploiement de l'opérateur Astra Control Center (`astra_control_center_operator_deploy.yaml`) pour faire référence à votre registre local et à votre secret.

```
vim astra_control_center_operator_deploy.yaml
```

ⓘ Un échantillon annoté YAML suit ces étapes.

a. 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
```

b. Changer `[your_registry_path]` pour le `kube-rbac-proxy` image dans le chemin du registre où vous avez poussé les images dans un étape précédente.

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

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```
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: [your_registry_path]/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: [your_registry_path]/acc-operator:[version x.y.z]
        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
```

3. Poser le conducteur du centre de commande Astra :

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

Exemple de réponse :

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
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
```

4. 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 :

<code>accountName</code>

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`

<code>astraVersion</code>

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	`22.11.0-82`

<code>astraAddress</code>

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`

<code>autoSupport</code>

Vos sélections dans cette section déterminent si vous allez participer à l'application de support proactif de NetApp, à NetApp Active IQ et à l'endroit où les données seront envoyées. 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`

<code>email</code>

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`

<code>firstName</code>

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`

<code>LastName</code>

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`

<code>imageRegistry</code>

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 dans lequel vous avez poussé les images dans le étape précédente. Ne pas utiliser `http://` ou `https://` dans le nom du registre.	chaîne	`example.registry.com/astra`
`imageRegistry.secret`	Obligatoire si la chaîne que vous avez entrée pour `imageRegistry.name' requires a secret.` `IMPORTANT: If you are using a registry that does not require authorization, you must delete this` `secret` ligne comprise entre `imageRegistry` sinon, l'installation échouera.	Nom du secret Kubernetes utilisé pour s'authentifier auprès du registre d'images.	chaîne	`astra-registry-cred`

<code>storageClass</code>

Réglage	Guidage	Type	Exemple
`storageClass`	Modifiez le `storageClass` valeur à partir de `ontap-gold` Sur une autre ressource de stockage Trident, tel que requis par votre installation. Lancer la commande `kubectl get sc` pour déterminer vos classes de stockage configurées existantes. L'une des classes de stockage basées sur Trident 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`

<code>volumeReclaimPolicy</code>

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`

<code>ingressType</code>

Réglage	Guidage	Type	Options
`ingressType`	Utilisez l'un des types d'entrées suivants :*`Generic`* (`ingressType: "Generic"`) (Par défaut) utilisez cette option lorsque vous avez un autre contrôleur d'entrée en service ou que vous préférez utiliser votre propre contrôleur d'entrée. Après le déploiement du centre de contrôle Astra, vous devez configurer le "contrôleur d'entrée" Pour exposer Astra Control Center avec une URL.**`AccTraefik`** (`ingressType: "AccTraefik"`) Utilisez cette option si vous préférez ne pas configurer un 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 voit attribuer une adresse IP externe accessible. Si des équilibreurs de charge sont autorisés dans votre environnement et que vous n'en avez pas encore configuré, vous pouvez utiliser MetalLB ou un autre équilibreur de charge de service externe pour attribuer une adresse IP externe au service. Dans la configuration du serveur	chaîne	• `Generic` (il s'agit de la valeur par défaut) • `AccTraefik`

<code>astraResourcesScaler</code>

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`

<code>crds</code>

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)

```
<strong>astra_control_center.yaml</strong>
```

```
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: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  astraResourcesScaler: "Default"
  additionalValues: {}
  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]
   ```

   Exemple de réponse :

   ```
   namespace/netapp-acc created
   ```

2. 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]
```

Exemple de réponse :

```
astracontrolcenter.astra.netapp.io/astra created
```

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 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.

Exemple de réponse

```
NAME                                          READY   STATUS
RESTARTS        AGE
acc-helm-repo-76d8d845c9-ggds2                1/1     Running   0
14m
activity-6cc67ff9f4-z48mr                     1/1     Running   2
(8m32s ago)   9m
api-token-authentication-7s67v                1/1     Running   0
8m56s
api-token-authentication-bplb4                1/1     Running   0
8m56s
api-token-authentication-p2c9z                1/1     Running   0
8m56s
asup-6cdfbc6795-md8vn                         1/1     Running   0
9m14s
authentication-9477567db-8hnc9                1/1     Running   0
7m4s
bucketservice-f4dbdfcd6-wqzkw                 1/1     Running   0
8m48s
cert-manager-bb756c7c4-wm2cv                  1/1     Running   0
14m
cert-manager-cainjector-c9bb86786-8wrf5       1/1     Running   0
14m
cert-manager-webhook-dd465db99-j2w4x          1/1     Running   0
14m
certificates-68dff9cdd6-kcvml                 1/1     Running   2
(8m43s ago)   9m2s
certificates-68dff9cdd6-rsnsb                 1/1     Running   0
9m2s
cloud-extension-69d48c956c-2s8dt              1/1     Running   3
(8m43s ago)   9m24s
cloud-insights-service-7c4f48b978-7gvlh       1/1     Running   3
(8m50s ago)   9m28s
composite-compute-7d9ff5f68-nxbhl             1/1     Running   0
8m51s
composite-volume-57b4756d64-nl66d             1/1     Running   0
9m13s
credentials-6dbc55f89f-qpzff                  1/1     Running   0
11m
entitlement-67bfb6d7-gl6kp                    1/1     Running   4
(8m33s ago)   9m38s
features-856cc4dccc-mxbdb                     1/1     Running   0
9m20s
fluent-bit-ds-4rtsp                           1/1     Running   0
6m54s
```

```
fluent-bit-ds-9rqll                          1/1     Running   0
6m54s
fluent-bit-ds-w5mp7                          1/1     Running   0
6m54s
graphql-server-7c7cc49776-jz2kn              1/1     Running   0
2m29s
identity-87c59c975-9jpnf                     1/1     Running   0
9m6s
influxdb2-0                                  1/1     Running   0
13m
keycloak-operator-84ff6d59d4-qcnmc           1/1     Running   0
7m1s
krakend-cbf6c7df9-mdtzv                      1/1     Running   0
2m30s
license-5b888b78bf-plj6j                     1/1     Running   0
9m32s
login-ui-846b4664dd-fz8hv                    1/1     Running   0
2m24s
loki-0                                       1/1     Running   0
13m
metrics-facade-779cc9774-n26rw               1/1     Running   0
9m18s
monitoring-operator-974db78f-pkspq           2/2     Running   0
6m58s
nats-0                                       1/1     Running   0
13m
nats-1                                       1/1     Running   0
13m
nats-2                                       1/1     Running   0
13m
nautilus-7bdc7ddc54-49tfn                    1/1     Running   0
7m50s
nautilus-7bdc7ddc54-cwc79                    1/1     Running   0
9m36s
openapi-5584ff9f46-gbrdj                     1/1     Running   0
9m17s
openapi-5584ff9f46-z9mzk                     1/1     Running   0
9m17s
packages-bfc58cc98-lpxq9                     1/1     Running   0
8m58s
polaris-consul-consul-server-0               1/1     Running   0
13m
polaris-consul-consul-server-1               1/1     Running   0
13m
polaris-consul-consul-server-2               1/1     Running   0
13m
```

```
polaris-keycloak-0                          1/1     Running   3
(6m15s ago)   6m56s
polaris-keycloak-1                          1/1     Running   0
4m22s
polaris-keycloak-2                          1/1     Running   0
3m41s
polaris-keycloak-db-0                       1/1     Running   0
6m56s
polaris-keycloak-db-1                       1/1     Running   0
4m23s
polaris-keycloak-db-2                       1/1     Running   0
3m36s
polaris-mongodb-0                           2/2     Running   0
13m
polaris-mongodb-1                           2/2     Running   0
13m
polaris-mongodb-2                           2/2     Running   0
12m
polaris-ui-5ccff47897-8rzgh                 1/1     Running   0
2m33s
polaris-vault-0                             1/1     Running   0
13m
polaris-vault-1                             1/1     Running   0
13m
polaris-vault-2                             1/1     Running   0
13m
public-metrics-6cb7bfc49b-p54xm             1/1     Running   1
(8m29s ago)   9m31s
storage-backend-metrics-5c77994586-kjn48    1/1     Running   0
8m52s
storage-provider-769fdc858c-62w54           1/1     Running   0
8m54s
task-service-9ffc484c5-kx9f4                1/1     Running   3
(8m44s ago)   9m34s
telegraf-ds-bphb9                           1/1     Running   0
6m54s
telegraf-ds-rtsm2                           1/1     Running   0
6m54s
telegraf-ds-s9h5h                           1/1     Running   0
6m54s
telegraf-rs-lbpv7                           1/1     Running   0
6m54s
telemetry-service-57cfb998db-zjx78          1/1     Running   1
(8m40s ago)   9m26s
tenancy-5d5dfbcf9f-vmbxh                    1/1     Running   0
9m5s
```

```
traefik-7b87c4c474-jmgp2                  1/1     Running   0
2m24s
traefik-7b87c4c474-t9k8x                  1/1     Running   0
2m24s
trident-svc-c78f5b6bd-nwdsq               1/1     Running   0
9m22s
vault-controller-55bbc96668-c6425         1/1     Running   0
11m
vault-controller-55bbc96668-lq9n9         1/1     Running   0
11m
vault-controller-55bbc96668-rfkgg         1/1     Running   0
11m
```

2. (Facultatif) pour vous assurer que l'installation est terminée, vous pouvez regarder le `acc-operator` journaux utilisant la commande suivante.

```
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.

3. 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 connectez 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   22.11.0-82  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`).

Après le déploiement du centre de contrôle Astra, vous devrez configurer le contrôleur d'entrée pour exposer le centre de contrôle Astra à 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 pour les types de contrôleurs d'entrée suivants :

- Entrée Istio
- Contrôleur d'entrée Nginx
- Contrôleur d'entrée OpenShift

Ce dont vous avez besoin

- 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 modifierez le mot de passe de l'administrateur par défaut et vous connecterez au tableau de bord de l'interface utilisateur de 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.

Étapes

1. 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
```

Et la suite

- (Facultatif) en fonction de votre environnement, effectuez l'installation complète après l'installation "étapes de configuration".
- Terminez le déploiement en effectuant le processus "tâches de configuration".

=
:allow-uri-read:

Informations sur le copyright

Informations sur les marques commerciales