



Installieren Sie Astra Control Center

Astra Control Center

NetApp
June 06, 2024

Inhalt

Installieren Sie das Astra Control Center mithilfe des Standardprozesses	1
Laden Sie das Astra Control Center herunter und extrahieren Sie es	2
Installieren Sie das NetApp Astra kubectl Plug-in	3
Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu	3
Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen	5
Installieren Sie den Operator Astra Control Center	7
Konfigurieren Sie Astra Control Center	10
Komplette Astra Control Center und Bedienerinstallation	23
Überprüfen Sie den Systemstatus	24
Eindringen für den Lastenausgleich einrichten	29
Melden Sie sich in der UI des Astra Control Center an	32
Beheben Sie die Fehlerbehebung für die Installation	32
Wie es weiter geht	33

Installieren Sie das Astra Control Center mithilfe des Standardprozesses

Laden Sie zum Installieren des Astra Control Center das Installationspaket von der NetApp Support Site herunter und führen Sie die folgenden Schritte aus. Mit diesem Verfahren können Sie Astra Control Center in Internet-angeschlossenen oder luftgekapselten Umgebungen installieren.

Andere Installationsverfahren

- **Installation mit RedHat OpenShift OperatorHub:** Verwenden Sie dies ["Alternativverfahren"](#) So installieren Sie Astra Control Center auf OpenShift mit OperatorHub.
- **In der öffentlichen Cloud mit Cloud Volumes ONTAP-Backend installieren:** Verwenden ["Derartige Verfahren"](#) Zur Installation von Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) oder Microsoft Azure mit einem Cloud Volumes ONTAP Storage-Back-End

Eine Demonstration des Installationsvorgangs für Astra Control Center finden Sie unter ["Dieses Video"](#).

Was Sie benötigen

- ["Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor"](#).
- Wenn Sie POD-Sicherheitsrichtlinien in Ihrer Umgebung konfiguriert haben oder konfigurieren möchten, sollten Sie sich mit den POD-Sicherheitsrichtlinien vertraut machen und wie diese sich auf die Installation des Astra Control Center auswirken. Siehe ["Einschränkungen der POD-Sicherheitsrichtlinie verstehen"](#).
- Stellen Sie sicher, dass alle API-Services in einem ordnungsgemäßen Zustand und verfügbar sind:

```
kubectl get apiservices
```

- Stellen Sie sicher, dass der Astra FQDN, den Sie verwenden möchten, für diesen Cluster routingfähig ist. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.
- Wenn bereits ein Zertifikat-Manager im Cluster vorhanden ist, müssen Sie einen Teil durchführen ["Erforderliche Schritte"](#) Damit Astra Control Center nicht versucht, seinen eigenen Cert Manager zu installieren. Standardmäßig installiert Astra Control Center während der Installation einen eigenen Cert-Manager.

Über diese Aufgabe

Mit dem Astra Control Center-Installationsprozess können Sie Folgendes tun:

- Installieren Sie die Astra-Komponenten im `netapp-acc` (Oder Name des benutzerdefinierten Namespace).
- Erstellen Sie ein Standard-Astra Control Owner-Administratorkonto.
- Legen Sie eine E-Mail-Adresse für einen Administrator und ein Standard-Kennwort für die Ersteinrichtung fest. Diesem Benutzer wird die Owner-Rolle zugewiesen, die für die erste Anmeldung bei der UI benötigt wird.

- Stellen Sie fest, dass alle Astra Control Center Pods ausgeführt werden.
- Installieren Sie die Astra Control Center-UI.



Löschen Sie den Operator Astra Control Center nicht (z. B. `kubectl delete -f astra_control_center_operator_deploy.yaml`) Zu jeder Zeit während der Astra Control Center Installation oder Betrieb, um das Löschen von Pods zu vermeiden.

Schritte

Gehen Sie wie folgt vor, um Astra Control Center zu installieren:

- [Laden Sie das Astra Control Center herunter und extrahieren Sie es](#)
- [Installieren Sie das NetApp Astra kubectl Plug-in](#)
- [Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu](#)
- [Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen](#)
- [Installieren Sie den Operator Astra Control Center](#)
- [Konfigurieren Sie Astra Control Center](#)
- [Komplette Astra Control Center und Bedienerinstallation](#)
- [Überprüfen Sie den Systemstatus](#)
- [Eindringen für den Lastenausgleich einrichten](#)
- [Melden Sie sich in der UI des Astra Control Center an](#)

Laden Sie das Astra Control Center herunter und extrahieren Sie es

1. Wechseln Sie zum "[Astra Control Center-Seite zum Herunterladen der Testversion](#)" Auf der NetApp Support Site
2. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`).
3. (Empfohlen, aber optional) Laden Sie das Zertifikaten- und Unterschriftenpaket für Astra Control Center herunter (`astra-control-center-certs-[version].tar.gz`) Um die Signatur des Pakets zu überprüfen:

```
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
```

Die Ausgabe wird angezeigt `verified OK` Nach erfolgreicher Überprüfung.

4. Extrahieren Sie die Bilder aus dem Astra Control Center Bundle:

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

Installieren Sie das NetApp Astra kubectl Plug-in

Das NetApp Astra kubectl Kommandozeilen-Plug-in spart Zeit, wenn es gängige Aufgaben im Zusammenhang mit der Bereitstellung und dem Upgrade des Astra Control Center ausführt.

Was Sie benötigen

NetApp bietet Plug-ins-Binärdateien für verschiedene CPU-Architekturen und Betriebssysteme. Sie müssen wissen, welche CPU und welches Betriebssystem Sie haben, bevor Sie diese Aufgabe ausführen.

Schritte

1. Geben Sie die verfügbaren Plug-ins-Binärdateien von NetApp Astra kubectl an und notieren Sie sich den Namen der für Ihr Betriebssystem und die CPU-Architektur erforderlichen Datei:



Die kubectl Plugin-Bibliothek ist Teil des tar-Bündels und wird in den Ordner extrahiert `kubectl-astra`.

```
ls kubectl-astra/
```

2. Verschieben Sie die richtige Binärdatei in den aktuellen Pfad, und benennen Sie sie in um `kubectl-astra`:

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

Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

1. Führen Sie die entsprechende Schrittfolge für Ihre Container-Engine durch:

Docker

1. Wechseln Sie in das Stammverzeichnis des Tarballs. Sie sollten diese Datei und das Verzeichnis sehen:

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

2. Übertragen Sie die Paketbilder im Astra Control Center-Bildverzeichnis in Ihre lokale Registrierung. Führen Sie die folgenden Ersetzungen durch, bevor Sie den ausführen `push-images` Befehl:

- Ersetzen Sie `<BUNDLE_FILE>` durch den Namen der Astra Control Bundle-Datei (`acc.manifest.bundle.yaml`).
- `<MY_FULL_REGISTRY_PATH>` durch die URL des Docker Repositorys ersetzen, beispielsweise "`<a href="https://<docker-registry>" class="bare">https://<docker-registry>"`".
- Ersetzen Sie `<MY_REGISTRY_USER>` durch den Benutzernamen.
- Ersetzen Sie `<MY_REGISTRY_TOKEN>` durch ein autorisiertes Token für die Registrierung.

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

Podman

1. Wechseln Sie in das Stammverzeichnis des Tarballs. Sie sollten diese Datei und das Verzeichnis sehen:

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

2. Melden Sie sich bei Ihrer Registrierung an:

```
podman login <YOUR_REGISTRY>
```

3. Vorbereiten und Ausführen eines der folgenden Skripts, das für die von Ihnen verwendete Podman-Version angepasst ist. Ersetzen Sie `<MY_FULL_REGISTRY_PATH>` durch die URL Ihres Repositorys, die alle Unterverzeichnisse enthält.

```
<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

```

Podman 3

```

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

```



Der Bildpfad, den das Skript erstellt, sollte abhängig von Ihrer Registrierungskonfiguration wie folgt aussehen:

<https://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version>

Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen

1. Exportieren Sie den KUBECONFIG für den Hostcluster Astra Control Center:

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



Bevor Sie die Installation abgeschlossen haben, stellen Sie sicher, dass Ihr KUBECONFIG auf den Cluster zeigt, in dem Sie Astra Control Center installieren möchten. Die KUBECONFIG kann nur einen Kontext enthalten.

2. Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, müssen Sie Folgendes tun:

a. Erstellen Sie die `netapp-acc-operator` Namespace:

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

Antwort:

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

b. Erstellen Sie ein Geheimnis für das `netapp-acc-operator` Namespace. Fügen Sie Docker-Informationen hinzu und führen Sie den folgenden Befehl aus:



Platzhalter `your_registry_path` Sollte die Position der Bilder, die Sie früher hochgeladen haben, entsprechen (z. B. `[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]
```

Beispielantwort:

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



Wenn Sie den Namespace löschen, nachdem das Geheimnis generiert wurde, erstellen Sie den Namespace neu und generieren Sie dann das Geheimnis für den Namespace neu.

c. Erstellen Sie die `netapp-acc` (Oder Name des benutzerdefinierten Namespace).

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

Beispielantwort:


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

- d. Erstellen Sie ein Geheimnis für das netapp-acc (Oder Name des benutzerdefinierten Namespace). Fügen Sie Docker-Informationen hinzu und führen Sie den folgenden Befehl aus:

```
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]
```

Antwort

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

Installieren Sie den Operator Astra Control Center

1. Telefonbuch ändern:

```
cd manifests
```

2. Bearbeiten Sie die YAML-Implementierung des Astra Control Center-Bediensers (astra_control_center_operator_deploy.yaml) Zu Ihrem lokalen Register und Geheimnis zu verweisen.

```
vim astra_control_center_operator_deploy.yaml
```



Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

- a. Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, ersetzen Sie die Standardzeile von imagePullSecrets: [] Mit folgenden Optionen:

```
imagePullSecrets:  
- name: astra-registry-cred
```

- b. Ändern [your_registry_path] Für das kube-rbac-proxy Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben [Vorheriger Schritt](#).
- c. Ändern [your_registry_path] Für das acc-operator-controller-manager Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben [Vorheriger Schritt](#).

```
<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. Installieren Sie den Astra Control Center-Operator:

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

Beispielantwort:

```
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. Überprüfen Sie, ob Pods ausgeführt werden:

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

Konfigurieren Sie Astra Control Center

1. Bearbeiten Sie die Datei Astra Control Center Custom Resource (CR) (`astra_control_center.yaml`)
Zur Berücksichtigung, Unterstützung, Registrierung und anderen notwendigen Konfigurationen:

```
vim astra_control_center.yaml
```



Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

2. Ändern oder bestätigen Sie die folgenden Einstellungen:

`<code>accountName</code>`

Einstellung	Anleitung	Typ	Beispiel
accountName	Ändern Sie das <code>accountName</code> Zeichenfolge an den Namen, den Sie dem Astra Control Center-Konto zuordnen möchten. Es kann nur ein AccountName geben.	Zeichenfolge	Example

`<code>astraVersion</code>`

Einstellung	Anleitung	Typ	Beispiel
astraVersion	Die zu implementierende Version des Astra Control Center: Für diese Einstellung ist keine Aktion erforderlich, da der Wert bereits ausgefüllt wird.	Zeichenfolge	22.11.0-82

`<code>astraAddress</code>`

Einstellung	Anleitung	Typ	Beispiel
astraAddress	<p>Ändern Sie das <code>astraAddress</code> Zeichenfolge an den FQDN (empfohlen) oder die IP-Adresse, die Sie in Ihrem Browser verwenden möchten, um auf Astra Control Center zuzugreifen. Diese Adresse legt fest, wie Astra Control Center in Ihrem Rechenzentrum zu finden ist und ist die gleiche FQDN- oder IP-Adresse, die Sie von Ihrem Load Balancer bereitgestellt haben, wenn Sie fertig sind "Anforderungen des Astra Control Centers". HINWEIS: Nicht verwenden <code>http://</code> Oder <code>https://</code> In der Adresse. Kopieren Sie diesen FQDN zur Verwendung in einem Später Schritt.</p>	Zeichenfolge	astra.example.com

<code>autoSupport</code>

Anhand Ihrer Auswahl in diesem Abschnitt wird bestimmt, ob Sie an der pro-aktiven Support-Applikation von NetApp, dem NetApp Active IQ und dem Sendeort von Daten teilnehmen. Eine Internetverbindung ist erforderlich (Port 442), und alle Supportdaten werden anonymisiert.

Einstellung	Nutzung	Anleitung	Typ	Beispiel
<code>autoSupport.enrolled</code>	Entweder <code>enrolled</code> Oder <code>url</code> Felder müssen ausgewählt werden	Ändern <code>enrolled</code> Für AutoSupport bis <code>false</code> Für Websites ohne Internetverbindung oder Aufbewahrung <code>true</code> Für verbundene Standorte. Eine Einstellung von <code>true</code> Anonyme Daten können zu Supportzwecken an NetApp gesendet werden. Die Standardwahl ist <code>false</code> Und zeigt an, dass keine Support-Daten an NetApp gesendet werden.	Boolesch	<code>false</code> (Dieser Wert ist der Standardwert)
<code>autoSupport.url</code>	Entweder <code>enrolled</code> Oder <code>url</code> Felder müssen ausgewählt werden	Diese URL legt fest, wo die anonymen Daten gesendet werden.	Zeichenfolge	https://support.netapp.com/asupprod/post/1.0/postAsup

`<code>email</code>`

Einstellung	Anleitung	Typ	Beispiel
<code>email</code>	Ändern Sie das <code>email</code> Zeichenfolge zur standardmäßigen ursprünglichen Administratoradresse. Kopieren Sie diese E-Mail-Adresse zur Verwendung in A Später Schritt . Diese E-Mail-Adresse wird als Benutzername für das erste Konto verwendet, um sich bei der UI anzumelden und wird über Ereignisse in Astra Control informiert.	Zeichenfolge	<code>admin@example.com</code>

`<code>firstName</code>`

Einstellung	Anleitung	Typ	Beispiel
<code>firstName</code>	Der erste Name des mit dem Astra-Konto verknüpften Standardadministrators. Der hier verwendete Name wird nach der ersten Anmeldung in einer Überschrift in der UI angezeigt.	Zeichenfolge	SRE

`<code>lastName</code>`

Einstellung	Anleitung	Typ	Beispiel
<code>lastName</code>	Der Nachname des mit dem Astra-Konto verknüpften Standard-Initialadministrators. Der hier verwendete Name wird nach der ersten Anmeldung in einer Überschrift in der UI angezeigt.	Zeichenfolge	Admin

<code>imageRegistry</code>

Ihre Auswahl in diesem Abschnitt definiert die Container-Image-Registry, die die Astra-Anwendungsabbilder, den Astra Control Center Operator und das Astra Control Center Helm Repository hostet.

Einstellung	Nutzung	Anleitung	Typ	Beispiel
<code>imageRegistry.name</code>	Erforderlich	Der Name der Bildregistrierung, in der Sie die Bilder in geschoben haben Vorheriger Schritt . Verwenden Sie es nicht <code>http://</code> Oder <code>https://</code> Im Registrierungsnamen.	Zeichenfolge	<code>example.registry.com/astra</code>
<code>imageRegistry.secret</code>	Erforderlich, wenn der von Ihnen eingegebene String eingegeben wird <code>imageRegistry.name</code> requires a secret. IMPORTANT: If you are using a registry that does not require authorization, you must delete this <code>secret</code> Zeile in <code>imageRegistry</code> Oder die Installation schlägt fehl.	Der Name des Kubernetes Secret, das zur Authentifizierung mit der Bildregistrierung verwendet wird.	Zeichenfolge	<code>astra-registry-cred</code>

`<code>storageClass</code>`

Einstellung	Anleitung	Typ	Beispiel
storageClass	<p>Ändern Sie das <code>storageClass</code> Wert von <code>ontap-gold</code> Bei Bedarf einer anderen Trident Storage Class Ressource verwenden. Führen Sie den Befehl aus <code>kubectl get sc</code> So ermitteln Sie Ihre vorhandenen konfigurierten Speicherklassen. Eine der Trident-basierten Speicherklassen muss in die Manifest-Datei eingegeben werden (<code>astra-control-center- <version>.manifest</code>) Und wird für Astra PVS verwendet. Wenn er nicht festgelegt ist, wird die Standard-Speicherklasse verwendet. HINWEIS: Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass diese die einzige Storage-Klasse mit der Standardbeschriftung ist.</p>	Zeichenfolge	ontap-gold

`<code>volumeReclaimPolicy</code>`

Einstellung	Anleitung	Typ	Optionen
volumeReclaimPolicy	Damit wird die Rückgewinnungsrichtlinie für die PVS von Astra festgelegt. Festlegen dieser Richtlinie auf Retain Behält persistente Volumes nach dem Löschen von Astra bei. Festlegen dieser Richtlinie auf Delete Löscht persistente Volumes nach dem Löschen von astra. Wenn dieser Wert nicht festgelegt ist, werden die PVS beibehalten.	Zeichenfolge	<ul style="list-style-type: none">• Retain (Dies ist der Standardwert)• Delete

`<code>ingressType</code>`





Einstellung	Anleitung	Typ	Optionen
ingressType	<p>Verwenden Sie einen der folgenden Eingangstypen: *Generic* (ingressType: "Generic") (Standard)</p> <p>Verwenden Sie diese Option, wenn Sie einen anderen Ingress-Controller verwenden oder Ihren eigenen Ingress-Controller verwenden möchten. Nach der Implementierung des Astra Control Center müssen Sie den konfigurieren "Eingangs-Controller"</p> <p>Um Astra Control Center mit einer URL zu zeigen. AccTraefik (ingressType: "AccTraefik")</p> <p>Verwenden Sie diese Option, wenn Sie keine Ingress-Controller konfigurieren möchten. Dies implementiert das Astra Control Center traefik Gateway als Service des Typs Kubernetes Load Balancer: Astra Control Center nutzt einen Service vom Typ „loadbalancer“ (svc/traefik Im Astra Control Center Namespace) und erfordert, dass ihm eine zugängliche externe IP-Adresse zugewiesen wird. Wenn in Ihrer Umgebung Load Balancer zugelassen sind und Sie noch keine konfiguriert haben, können Sie MetallB oder einen anderen externen Service Load Balancer verwenden, um dem Dienst eine externe IP-Adresse zuzuweisen. In der</p>	Zeichenfolge	<ul style="list-style-type: none"> • Generic (Dies ist der Standardwert) • AccTraefik

`<code>astraResourcesScaler</code>`

Einstellung	Anleitung	Typ	Optionen
<code>astraResourcesScaler</code>	<p>Skalierungsoptionen für die Ressourcengrenzen von AstraControlCenter. Astra Control Center implementiert standardmäßig mit Ressourcenanfragen, die für die meisten Komponenten in Astra bereitgestellt werden. Mit dieser Konfiguration verbessert sich die Leistung des Astra Control Center Software-Stacks auch bei erhöhter Applikationslast und -Skalierung. In Szenarien mit kleineren Entwicklungs- oder Testclustern jedoch das CR-Feld <code>astraResourcesScaler</code> Kann auf festgelegt werden <code>Off</code>. Dadurch werden Ressourcenanforderungen deaktiviert und die Bereitstellung auf kleineren Clustern ist möglich.</p>	Zeichenfolge	<ul style="list-style-type: none">• Default (Dies ist der Standardwert)• Off

`<code>crds</code>`

Ihre Auswahl in diesem Abschnitt legt fest, wie Astra Control Center mit CRDs umgehen soll.

Einstellung	Anleitung	Typ	Beispiel
<code>crds.externalCertManager</code>	Wenn Sie einen externen Zertifikaten-Manager verwenden, ändern Sie <code>externalCertManager</code> Bis <code>true</code> . Der Standardwert <code>false</code> Führt dazu, dass Astra Control Center während der Installation seine eigenen CRT-Manager-CRDs installiert. CRDs sind Cluster-weite Objekte, die sich auf andere Teile des Clusters auswirken können. Mit diesem Flag können Sie dem Astra Control Center signalisieren, dass diese CRDs vom Clusteradministrator außerhalb des Astra Control Center installiert und verwaltet werden.	Boolesch	<code>False</code> (Dieser Wert ist der Standardwert)
<code>crds.externalTraffic</code>	Astra Control Center installiert standardmäßig die erforderlichen Trafik-CRDs. CRDs sind Cluster-weite Objekte, die sich auf andere Teile des Clusters auswirken können. Mit diesem Flag können Sie dem Astra Control Center signalisieren, dass diese CRDs vom Clusteradministrator außerhalb des Astra Control Center installiert und verwaltet werden.	Boolesch	<code>False</code> (Dieser Wert ist der Standardwert)

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

Komplette Astra Control Center und Bedienerinstallation

1. Wenn Sie dies in einem vorherigen Schritt nicht bereits getan haben, erstellen Sie das `netapp-acc` (Oder benutzerdefinierter) Namespace:

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

Beispielantwort:

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

2. Installieren Sie das Astra Control Center im `netapp-acc` (Oder Ihr individueller) Namespace:

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

Beispielantwort:

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

Überprüfen Sie den Systemstatus

Sie können den Systemstatus mithilfe von kubectl-Befehlen überprüfen. Wenn Sie OpenShift verwenden möchten, können Sie vergleichbare oc-Befehle für Verifizierungsschritte verwenden.

Schritte

1. Vergewissern Sie sich, dass alle Systemkomponenten erfolgreich installiert wurden.

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

Jeder Pod sollte einen Status von `Running` haben. Es kann mehrere Minuten dauern, bis die System-Pods implementiert sind.

Beispielantwort

NAME	READY	STATUS	
RESTARTS	AGE		
acc-helm-repo-76d8d845c9-ggds2 14m	1/1	Running	0
activity-6cc67ff9f4-z48mr (8m32s ago) 9m	1/1	Running	2
api-token-authentication-7s67v 8m56s	1/1	Running	0
api-token-authentication-bplb4 8m56s	1/1	Running	0
api-token-authentication-p2c9z 8m56s	1/1	Running	0
asup-6cdfbc6795-md8vn 9m14s	1/1	Running	0
authentication-9477567db-8hnc9 7m4s	1/1	Running	0
bucket-service-f4dbdfcd6-wqzkw 8m48s	1/1	Running	0
cert-manager-bb756c7c4-wm2cv 14m	1/1	Running	0
cert-manager-cainjector-c9bb86786-8wrf5 14m	1/1	Running	0
cert-manager-webhook-dd465db99-j2w4x 14m	1/1	Running	0
certificates-68dff9cdd6-kcvml (8m43s ago) 9m2s	1/1	Running	2
certificates-68dff9cdd6-rsnsb 9m2s	1/1	Running	0
cloud-extension-69d48c956c-2s8dt (8m43s ago) 9m24s	1/1	Running	3
cloud-insights-service-7c4f48b978-7gvlh (8m50s ago) 9m28s	1/1	Running	3
composite-compute-7d9ff5f68-nxbhl 8m51s	1/1	Running	0
composite-volume-57b4756d64-nl66d 9m13s	1/1	Running	0
credentials-6dbc55f89f-qpzff 11m	1/1	Running	0
entitlement-67fb6d7-gl6kp (8m33s ago) 9m38s	1/1	Running	4
features-856cc4dccc-mxbdb 9m20s	1/1	Running	0
fluent-bit-ds-4rtsp 6m54s	1/1	Running	0

fluent-bit-ds-9rql1	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 (6m15s ago) 6m56s	1/1	Running	3
polaris-keycloak-1 4m22s	1/1	Running	0
polaris-keycloak-2 3m41s	1/1	Running	0
polaris-keycloak-db-0 6m56s	1/1	Running	0
polaris-keycloak-db-1 4m23s	1/1	Running	0
polaris-keycloak-db-2 3m36s	1/1	Running	0
polaris-mongodb-0 13m	2/2	Running	0
polaris-mongodb-1 13m	2/2	Running	0
polaris-mongodb-2 12m	2/2	Running	0
polaris-ui-5ccff47897-8rzgh 2m33s	1/1	Running	0
polaris-vault-0 13m	1/1	Running	0
polaris-vault-1 13m	1/1	Running	0
polaris-vault-2 13m	1/1	Running	0
public-metrics-6cb7bfc49b-p54xm (8m29s ago) 9m31s	1/1	Running	1
storage-backend-metrics-5c77994586-kjn48 8m52s	1/1	Running	0
storage-provider-769fdc858c-62w54 8m54s	1/1	Running	0
task-service-9ffc484c5-kx9f4 (8m44s ago) 9m34s	1/1	Running	3
telegraf-ds-bphb9 6m54s	1/1	Running	0
telegraf-ds-rtsm2 6m54s	1/1	Running	0
telegraf-ds-s9h5h 6m54s	1/1	Running	0
telegraf-rs-lbpv7 6m54s	1/1	Running	0
telemetry-service-57cfb998db-zjx78 (8m40s ago) 9m26s	1/1	Running	1
tenancy-5d5dfbcf9f-vmboxh 9m5s	1/1	Running	0

```

traefik-7b87c4c474-jmcp2      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. (Optional) um sicherzustellen, dass die Installation abgeschlossen ist, können Sie sich die ansehen `acc-operator` Protokolle mit dem folgenden Befehl

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



`accHost` Die Cluster-Registrierung ist einer der letzten Vorgänge, und bei Ausfall wird die Implementierung nicht fehlschlagen. Sollten in den Protokollen ein Fehler bei der Cluster-Registrierung angegeben sein, können Sie die Registrierung erneut über das versuchen ["Fügen Sie in der UI einen Cluster-Workflow hinzu"](#) Oder API.

3. Wenn alle Pods ausgeführt werden, überprüfen Sie, ob die Installation erfolgreich war (`READY` Ist `True`) Und holen Sie sich das erste Setup-Passwort, das Sie verwenden, wenn Sie sich bei Astra Control Center:

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

Antwort:

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	22.11.0-82	10.111.111.111
True			



Den UUID-Wert kopieren. Das Passwort lautet `ACC-` Anschließend der UUID-Wert (`ACC-[UUID]`) Oder in diesem Beispiel `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`.

Eindringen für den Lastenausgleich einrichten

Sie können einen Kubernetes Ingress-Controller einrichten, der den externen Zugriff auf Services managt. Diese Verfahren enthalten Setup-Beispiele für einen Ingress-Controller, wenn Sie die Standardeinstellung von verwenden `ingressType: "Generic"` In der Astra Control Center Custom Resource (`astra_control_center.yaml`). Sie müssen diesen Vorgang nicht verwenden, wenn Sie angegeben haben `ingressType: "AccTraefik"` In der Astra Control Center Custom Resource (`astra_control_center.yaml`).

Nachdem Astra Control Center bereitgestellt wurde, müssen Sie den Ingress-Controller so konfigurieren, dass Astra Control Center mit einer URL verfügbar ist.

Die Einstellungsschritte unterscheiden sich je nach Typ des Ingress-Controllers. Astra Control Center unterstützt viele Ingress-Controller-Typen. Diese Einstellungsverfahren enthalten Beispielschritte für die folgenden Ingress-Controller-Typen:

- Istio Ingress
- Nginx-Ingress-Controller
- OpenShift-Eingangs-Controller

Was Sie benötigen

- Erforderlich **"Eingangs-Controller"** Sollte bereits eingesetzt werden.
- Der **"Eingangsklasse"** Entsprechend der Eingangs-Steuerung sollte bereits erstellt werden.

Schritte für Istio Ingress

1. Konfigurieren Sie Istio Ingress.



Bei diesem Verfahren wird davon ausgegangen, dass Istio mithilfe des Konfigurationsprofils „Standard“ bereitgestellt wird.

2. Sammeln oder erstellen Sie die gewünschte Zertifikatdatei und die private Schlüsseldatei für das Ingress Gateway.

Sie können ein CA-signiertes oder selbstsigniertes Zertifikat verwenden. Der allgemeine Name muss die Astra-Adresse (FQDN) sein.

Beispielbefehl:

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

3. Erstellen Sie ein Geheimnis `tls secret name` Vom Typ `kubernetes.io/tls` Für einen privaten TLS-Schlüssel und ein Zertifikat im `istio-system` namespace Wie in TLS Secrets beschrieben.

Beispielbefehl:

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



Der Name des Geheimnisses sollte mit dem übereinstimmen `spec.tls.secretName`
Verfügbar in `istio-ingress.yaml` Datei:

4. Bereitstellung einer Ingress-Ressource im `netapp-acc` (Oder Custom-Name) Namespace unter Verwendung des v1-Ressourcentyps für ein Schema (`istio-Ingress.yaml` Wird in diesem Beispiel verwendet):

```
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. Übernehmen Sie die Änderungen:

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

6. Überprüfen Sie den Status des Eingangs:


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

Antwort:

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

7. Astra Control Center-Installation abschließen.

Schritte für Nginx Ingress Controller

1. Erstellen Sie ein Geheimnis des Typs `kubernetes.io/tls` Für einen privaten TLS-Schlüssel und ein Zertifikat in `netapp-acc` (Oder Custom-Name) Namespace wie in beschrieben "[TLS-Geheimnisse](#)".
2. Bereitstellung einer Ingress-Ressource in `netapp-acc` (Oder Custom-Name) Namespace unter Verwendung des `v1`-Ressourcentyps für ein Schema (`nginx-Ingress.yaml` Wird in diesem Beispiel verwendet):

```
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. Übernehmen Sie die Änderungen:

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



NetApp empfiehlt die Installation des nginx Controllers als Bereitstellung statt als a daemonSet.

Schritte für OpenShift-Eingangs-Controller

1. Beschaffen Sie Ihr Zertifikat, und holen Sie sich die Schlüssel-, Zertifikat- und CA-Dateien für die OpenShift-Route bereit.
2. Erstellen Sie die OpenShift-Route:

```
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
```

Melden Sie sich in der UI des Astra Control Center an

Nach der Installation von Astra Control Center ändern Sie das Passwort für den Standardadministrator und melden sich im Astra Control Center UI Dashboard an.

Schritte

1. Geben Sie in einem Browser den FQDN ein (einschließlich `https://` Präfix), die Sie in verwendet haben `astraAddress` in `astra_control_center.yaml` CR, wenn [Sie haben das Astra Control Center installiert](#).
2. Akzeptieren Sie die selbstsignierten Zertifikate, wenn Sie dazu aufgefordert werden.



Sie können nach der Anmeldung ein benutzerdefiniertes Zertifikat erstellen.

3. Geben Sie auf der Anmeldeseite des Astra Control Center den Wert ein, den Sie für verwendet haben `email` in `astra_control_center.yaml` CR, wenn [Sie haben das Astra Control Center installiert](#), Gefolgt von dem anfänglichen Setup-Passwort (`ACC-[UUID]`).



Wenn Sie dreimal ein falsches Passwort eingeben, wird das Administratorkonto 15 Minuten lang gesperrt.

4. Wählen Sie **Login**.
5. Ändern Sie das Passwort, wenn Sie dazu aufgefordert werden.



Wenn dies Ihre erste Anmeldung ist und Sie das Passwort vergessen haben und noch keine anderen administrativen Benutzerkonten erstellt wurden, kontaktieren Sie ["NetApp Support"](#) Für Unterstützung bei der Kennwortwiederherstellung.

6. (Optional) Entfernen Sie das vorhandene selbst signierte TLS-Zertifikat und ersetzen Sie es durch ein ["Benutzerdefiniertes TLS-Zertifikat, signiert von einer Zertifizierungsstelle \(CA\)"](#).

Beheben Sie die Fehlerbehebung für die Installation

Wenn einer der Dienstleistungen in ist `ERROR` Status, können Sie die Protokolle überprüfen. Suchen Sie nach API-Antwortcodes im Bereich von 400 bis 500. Diese geben den Ort an, an dem ein Fehler aufgetreten ist.

Schritte

1. Um die Bedienerprotokolle des Astra Control Center zu überprüfen, geben Sie Folgendes ein:

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

Wie es weiter geht

- (Optional) Verarbeiten Sie abhängig von Ihrer Umgebung nach der Installation vollständig ["Konfigurationsschritte"](#).
- Führen Sie die Implementierung durch ["Setup-Aufgaben"](#).

=

:allow-uri-read:

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFT SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.