Übersicht über die Installation

Astra Control Center

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/de-de/astra-control-center-2310/get-started/cert-manager-prereqs.html on August 11, 2025. Always check docs.netapp.com for the latest.

Inhalt

Übersicht über die Installation . . . 1
- Installieren Sie das Astra Control Center mithilfe des Standardprozesses . . . 1
  - Laden Sie das Astra Control Center herunter und extrahieren Sie es . . . 4
  - Installieren Sie das NetApp Astra kubectl Plug-in . . . 5
  - Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu . . . 6
  - Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen . . . 8
  - Installieren Sie den Operator Astra Control Center . . . 10
  - Konfigurieren Sie Astra Control Center . . . 13
  - Komplette Astra Control Center und Bedienerinstallation . . . 28
  - Überprüfen Sie den Systemstatus . . . 29
  - Eindringen für den Lastenausgleich einrichten . . . 35
  - Melden Sie sich in der UI des Astra Control Center an . . . 39
  - Beheben Sie die Fehlerbehebung für die Installation . . . 39
  - Wie es weiter geht . . . 40
  - Konfigurieren Sie einen externen Zertifikaten-Manager . . . 40
- Installieren Sie Astra Control Center mit OpenShift OperatorHub . . . 42
  - Laden Sie das Astra Control Center herunter und extrahieren Sie es . . . 44
  - Installieren Sie das NetApp Astra kubectl Plug-in . . . 45
  - Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu . . . 46
  - Suchen Sie die Installationsseite des Bedieners . . . 48
  - Installieren Sie den Operator . . . 50
  - Installieren Sie Astra Control Center . . . 50
  - Erstellen Sie einen Registrierungsschlüssel . . . 52
  - Wie es weiter geht . . . 52
- Installieren Sie Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend . . . 52
  - Implementieren Sie Astra Control Center in Amazon Web Services . . . 53
  - Implementieren Sie Astra Control Center in der Google Cloud Platform . . . 57
  - Implementieren Sie Astra Control Center in Microsoft Azure . . . 62
- Konfigurieren Sie nach der Installation das Astra Control Center . . . 68
  - Ressourceneinschränkungen entfernen . . . 69
  - Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu . . . 70

Übersicht über die Installation

Wählen Sie einen der folgenden Astra Control Center-Installationsverfahren aus:

- "Installieren Sie das Astra Control Center mithilfe des Standardprozesses"
- "(Wenn Sie Red hat OpenShift verwenden) installieren Sie Astra Control Center mit OpenShift OperatorHub"
- "Installieren Sie Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend"

Je nach Umgebung kann nach der Installation des Astra Control Center eine zusätzliche Konfiguration erforderlich sein:

- "Konfigurieren Sie nach der Installation das Astra Control Center"

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 luftgekapderten Umgebungen installieren.

Für andere Installationsverfahren erweitern

- **Installation mit Red hat OpenShift OperatorHub**: Verwenden Sie diese "Alternativverfahren" So installieren Sie Astra Control Center unter Verwendung von OperatorHub auf OpenShift.
- **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".

Bevor Sie beginnen

- **Umweltvoraussetzungen erfüllen**: "Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor".

  Astra Control Center kann in einer dritten Fehlerdomäne oder an einem sekundären Standort implementiert werden. Dies wird für Applikationsreplizierung und nahtlose Disaster Recovery empfohlen.

- **Gesunde Dienste sicherstellen**: Überprüfen Sie, ob alle API-Dienste in einem gesunden Zustand sind und verfügbar sind:

```
kubectl get apiservices
```

- **Stellen Sie einen routingfähigen FQDN sicher**: Der Astra FQDN, den Sie verwenden möchten, kann zum Cluster weitergeleitet werden. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen

DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.

- **Configure cert Manager**: Wenn ein cert Manager bereits im Cluster existiert, müssen Sie einige 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.
- **Zugriff auf die NetApp Astra Control Image Registry**:
  Sie haben die Möglichkeit, Installations-Images und Funktionserweiterungen für Astra Control, wie z. B. Astra Control Provisioner, aus der NetApp-Image-Registrierung zu beziehen.

  Für Schritte erweitern

  a. Notieren Sie Ihre Astra Control Account-ID, die Sie zur Anmeldung in der Registrierung benötigen.

     Ihre Konto-ID wird in der Web-UI des Astra Control Service angezeigt. Wählen Sie das Symbol oben rechts auf der Seite aus, wählen Sie **API Access** aus und notieren Sie sich Ihre Konto-ID.

  b. Wählen Sie auf derselben Seite **API-Token generieren** aus und kopieren Sie die API-Token-Zeichenfolge in die Zwischenablage und speichern Sie sie in Ihrem Editor.

  c. Melden Sie sich in der Astra Control Registry an:

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

- **Betrachten Sie ein Service-Mesh**: Es wird dringend empfohlen, die Kommunikationskanäle des Astra Control-Host-Clusters mit einem zu sichern "Unterstütztes Service-Mesh".

Istio Service Mesh-Details

Für die Nutzung von Istio Service Mesh müssen Sie Folgendes tun:

- Fügen Sie ein hinzu `istio-injection:enabled` Etikett In den Astra Namespace vor der Implementierung von Astra Control Center.
- Verwenden Sie die `Generic` Einstellung für Eindringen Und bieten eine alternative Ingress für Externe Lastverteilung.
- Für Red hat OpenShift-Cluster müssen Sie definieren `NetworkAttachmentDefinition` In allen zugehörigen Astra Control Center-Namespaces (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` Für Anwendungscluster oder alle benutzerdefinierten Namespaces, die ersetzt wurden).

```
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
```

- **Nur ONTAP-SAN-Treiber**: Wenn Sie einen ONTAP-SAN-Treiber verwenden, stellen Sie sicher, dass Multipath auf allen Kubernetes-Clustern aktiviert ist.

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

> 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.

Laden Sie das Astra Control Center herunter und extrahieren Sie es

Sie können das Bundle von Astra Control Center von der NetApp Support-Website herunterladen oder das Bundle mithilfe von Docker aus der Image-Registrierung des Astra Control Service abrufen.

NetApp Support Website

1. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`) Vom "Download-Seite für Astra Control Center".
2. (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 Bündels zu überprüfen.

   Erweitern Sie, um Details anzuzeigen

   ```
   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.

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

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

Astra Control-Image-Registrierung

1. Melden Sie sich beim Astra Control Service an.
2. Wählen Sie im Dashboard **Deploy a self-Managed Instance of Astra Control** aus.
3. Folgen Sie den Anweisungen, um sich bei der Astra Control-Image-Registrierung anzumelden, das Astra Control Center-Installationsabbild zu ziehen und das Image zu extrahieren.

Installieren Sie das NetApp Astra kubectl Plug-in

Sie können das NetApp Astra kubectl Befehlszeilenschnittstelle-Plug-in verwenden, um Images in ein lokales Docker Repository zu verschieben.

Bevor Sie beginnen

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.

Wenn Sie das Plugin bereits von einer früheren Installation installiert haben, "Stellen Sie sicher, dass Sie über die neueste Version verfügen" Bevor Sie diese Schritte ausführen.

Schritte

1. Listen Sie die verfügbaren NetApp Astra kubectl Plugin-Binärdateien auf:

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

```
ls kubectl-astra/
```

2. Verschieben Sie die für Ihr Betriebssystem und die CPU-Architektur benötigte Datei 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 den sehen `acc.manifest.bundle.yaml` Datei und diese Verzeichnisse:

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

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

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=23.10.0-68
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=23.10.0-68
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://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/23.10.0-68/image:version
```

Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen

1. Exportieren Sie den kubeconfig für den Host-Cluster Astra Control Center:

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

Bevor Sie die Installation abschließen, vergewissern Sie sich, dass Ihr kubeconfig auf den Cluster zeigt, in dem Sie Astra Control Center installieren möchten.

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

Für Schritte erweitern

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

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

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/23.10.0-68`).

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

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]
```

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]
```

Installieren Sie den Operator Astra Control Center

1. Telefonbuch ändern:

```
cd manifests
```

2. Bearbeiten Sie die YAML-Implementierung des Astra Control Center-Bedieners (`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 `ASTRA_IMAGE_REGISTRY` Für das `kube-rbac-proxy` Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben Vorheriger Schritt.

c. Ändern `ASTRA_IMAGE_REGISTRY` Für das `acc-operator-controller-manager` Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben Vorheriger Schritt.

Erweitern für Beispiel astra_control_Center_Operator_deploy.yaml

```
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:23.10.72
        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
```

Erweitern für Probenantwort:

```
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
```

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 `accountName` 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	`23.10.0-68`

<code>astraAddress</code>

Einstellung	Anleitung	Typ	Beispiel
`astraAddress`	Ändern Sie das `astraAddress` 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 `http://` Oder `https://` In der Adresse. Kopieren Sie diesen FQDN zur Verwendung in einem Später Schritt.	Zeichenfolge	`astra.example.com`

<code>autoSupport</code>

Ihre Auswahl in diesem Abschnitt bestimmt, ob Sie an der proaktiven Support-Anwendung Digital Advisor von NetApp teilnehmen und an welchem Ort Daten gesendet werden. Eine Internetverbindung ist erforderlich (Port 442), und alle Supportdaten werden anonymisiert.

Einstellung	Nutzung	Anleitung	Typ	Beispiel
`autoSupport.enrolled`	Entweder `enrolled` Oder `url` Felder müssen ausgewählt werden	Ändern `enrolled` Für AutoSupport bis `false` Für Websites ohne Internetverbindung oder Aufbewahrung `true` Für verbundene Standorte. Eine Einstellung von `true` Ermöglicht das Senden anonymer Daten an NetApp zu Supportzwecken. Die Standardwahl ist `false` Und zeigt an, dass keine Support-Daten an NetApp gesendet werden.	Boolesch	`false` (Dieser Wert ist der Standardwert)
`autoSupport.url`	Entweder `enrolled` Oder `url` 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
`email`	Ändern Sie das `email` 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	`admin@example.com`

<code>firstName</code>

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

<code>storageClass</code>

Einstellung	Anleitung	Typ	Beispiel
`storageClass`	Ändern Sie das `storageClass` Wert von `ontap-gold` Je nach Installationsanforderungen zu einer anderen Ressource für Astra Trident Storage Class wechseln. Führen Sie den Befehl aus `kubectl get sc` So ermitteln Sie Ihre vorhandenen konfigurierten Speicherklassen. In die Manifest-Datei muss eine der Astra Trident-basierten Storage-Klassen eingegeben werden (`astra-control-center-<version>.manifest`) 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.	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	• `Retain` (Dies ist der Standardwert) • `Delete`

<code>ingressType</code>

Einstellung	Anleitung	Typ	Optionen
`ingressType`	Verwenden Sie einen der folgenden Eingangstypen: `Generic`* (`ingressType: "Generic"`) (Standard) 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" Um Astra Control Center mit einer URL zu zeigen. WICHTIG: Wenn Sie ein Service-Mesh mit Astra Control Center verwenden möchten, müssen Sie auswählen `Generic` Als Eindringen Typ und richten Sie Ihre eigenen "Eingangs-Controller". **`AccTraefik`** (`ingressType: "AccTraefik"`) Verwenden Sie diese Option, wenn Sie keinen 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	Zeichenfolge	• `Generic` (Dies ist der Standardwert) • `AccTraefik`

<code>scaleSize</code>

Einstellung	Anleitung	Typ	Optionen
`scaleSize`	Astra verwendet standardmäßig High Availability (HA). `scaleSize` Von `Medium`, Die die meisten Dienste in HA bereitstellt und mehrere Replikate für Redundanz bereitstellt. Mit `scaleSize` Als `Small`, Astra wird die Anzahl der Replikate für alle Dienste reduzieren, außer für wesentliche Dienste, um den Verbrauch zu reduzieren. TIPP: `Medium` Implementierungen bestehen aus etwa 100 Pods (einschließlich transienter Workloads). 100 Pods basieren auf drei Master Nodes und einer Konfiguration mit drei Worker Nodes). Beachten Sie die Einschränkungen bei der Netzwerkgrenze pro Pod, die in Ihrer Umgebung möglicherweise ein Problem darstellen, insbesondere bei der Betrachtung von Disaster-Recovery-Szenarien.	Zeichenfolge	• `Small` • `Medium` (Dies ist der Standardwert)

<code>astraResourcesScaler</code>

Einstellung	Anleitung	Typ	Optionen
`astraResourcesScaler`	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 `astraResourcesScalar` Kann auf festgelegt werden `Off`. Dadurch werden Ressourcenanforderungen deaktiviert und die Bereitstellung auf kleineren Clustern ist möglich.	Zeichenfolge	• `Default` (Dies ist der Standardwert) • `Off`

<code>additionalValues</code>

Fügen Sie dem Astra Control Center CR die folgenden zusätzlichen Werte hinzu, um ein bekanntes Problem bei der Installation zu vermeiden:

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

- Für die Kommunikation zwischen Astral Control Center und Cloud Insights ist die Überprüfung des TLS-Zertifikats standardmäßig deaktiviert. Sie können die TLS-Zertifizierungsüberprüfung für die Kommunikation zwischen Cloud Insights und dem Astra Control Center Host-Cluster und dem verwalteten Cluster aktivieren, indem Sie den folgenden Abschnitt in hinzufügen `additionalValues`.

```
 additionalValues:
   netapp-monitoring-operator:
     config:
       ciSkipTlsVerify: false
   cloud-insights-service:
     config:
       ciSkipTlsVerify: false
   telemetry-service:
     config:
       ciSkipTlsVerify: false
```

<code>crds</code>

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

Einstellung	Anleitung	Typ	Beispiel
`crds.externalCertManager`	Wenn Sie einen externen Zertifikaten-Manager verwenden, ändern Sie `externalCertManager` Bis `true`. Der Standardwert `false` 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	`False` (Dieser Wert ist der Standardwert)
`crds.externalTraefik`	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	`False` (Dieser Wert ist der Standardwert)

Stellen Sie sicher, dass Sie die richtige Storage-Klasse und den richtigen Ingress-Typ für Ihre Konfiguration ausgewählt haben, bevor Sie die Installation abschließen.

Erweitern für Beispiel 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"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  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]
```

2. Wenn Sie ein Service-Mesh mit Astra Control Center verwenden, fügen Sie dem die folgende Beschriftung hinzu `netapp-acc` Oder benutzerdefinierter Namespace:

> Ihre Art des Eingangs (`ingressType`) Muss auf gesetzt werden `Generic` Im Astra Control Center CR, bevor Sie mit diesem Befehl fortfahren.

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

3. (Empfohlen) "Aktivieren Sie strenge MTLs" Für Istio Service Mesh:

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

4. 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]
```

> Der Fahrer des Astra Control Center überprüft automatisch die Umgebungsanforderungen. Fehlt "Anforderungen" Kann dazu führen, dass Ihre Installation fehlschlägt oder Astra Control Center nicht ordnungsgemäß funktioniert. Siehe Nächster Abschnitt So prüfen Sie, ob Warnmeldungen zur automatischen Systemprüfung vorliegen.

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 beim Installationsprozess keine Warnmeldungen zu den Validierungsprüfungen ausgegeben wurden:

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

Zusätzliche Warnmeldungen werden auch in den Bedienerprotokollen des Astra Control Centers gemeldet.

2. Beheben Sie alle Probleme mit Ihrer Umgebung, die durch automatisierte Anforderungsprüfungen gemeldet wurden.

Sie können Probleme beheben, indem Sie sicherstellen, dass Ihre Umgebung den erfüllt "Anforderungen" Für Astra Control Center.

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

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

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

Erweitern, um die Probenantwort zu erhalten

```
NAME                                          READY   STATUS
RESTARTS     AGE
acc-helm-repo-6cc7696d8f-pmhm8                1/1     Running     0
9h
activity-597fb656dc-5rd4l                     1/1     Running     0
9h
activity-597fb656dc-mqmcw                     1/1     Running     0
9h
api-token-authentication-62f84                1/1     Running     0
9h
api-token-authentication-68nlf                1/1     Running     0
9h
api-token-authentication-ztgrm                1/1     Running     0
9h
asup-669d4ddbc4-fnmwp                         1/1     Running     1
(9h ago)   9h
authentication-78789d7549-lk686               1/1     Running     0
9h
bucketservice-65c7d95496-24x7l                1/1     Running     3
(9h ago)   9h
cert-manager-c9f9fbf9f-k8zq2                  1/1     Running     0
9h
cert-manager-c9f9fbf9f-qjlzm                  1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-b5qll       1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-p5whs       1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-4722b         1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-86kv5         1/1     Running     0
9h
certificates-59d9f6f4bd-2j899                 1/1     Running     0
9h
certificates-59d9f6f4bd-9d9k6                 1/1     Running     0
9h
certificates-expiry-check-28011180--1-8lkxz   0/1     Completed   0
9h
cloud-extension-5c9c9958f8-jdhrp              1/1     Running     0
9h
cloud-insights-service-5cdd5f7f-pp8r5         1/1     Running     0
9h
composite-compute-66585789f4-hxn5w            1/1     Running     0
9h
```

```
composite-volume-68649f68fd-tb7p4             1/1     Running     0
9h
credentials-dfc844c57-jsx92                   1/1     Running     0
9h
credentials-dfc844c57-xw26s                   1/1     Running     0
9h
entitlement-7b47769b87-4jb6c                  1/1     Running     0
9h
features-854d8444cc-c24b7                     1/1     Running     0
9h
features-854d8444cc-dv6sm                     1/1     Running     0
9h
fluent-bit-ds-9tlv4                           1/1     Running     0
9h
fluent-bit-ds-bpkcb                           1/1     Running     0
9h
fluent-bit-ds-cxmwx                           1/1     Running     0
9h
fluent-bit-ds-jgnhc                           1/1     Running     0
9h
fluent-bit-ds-vtr6k                           1/1     Running     0
9h
fluent-bit-ds-vxqd5                           1/1     Running     0
9h
graphql-server-7d4b9d44d5-zdbf5               1/1     Running     0
9h
identity-6655c48769-4pwk8                     1/1     Running     0
9h
influxdb2-0                                   1/1     Running     0
9h
keycloak-operator-55479d6fc6-slvmt            1/1     Running     0
9h
krakend-f487cb465-78679                       1/1     Running     0
9h
krakend-f487cb465-rjsxx                       1/1     Running     0
9h
license-64cbc7cd9c-qxsr8                      1/1     Running     0
9h
login-ui-5db89b5589-ndb96                     1/1     Running     0
9h
loki-0                                        1/1     Running     0
9h
metrics-facade-8446f64c94-x8h7b               1/1     Running     0
9h
monitoring-operator-6b44586965-pvcl4          2/2     Running     0
9h
```

```
nats-0                                          1/1     Running     0
9h
nats-1                                          1/1     Running     0
9h
nats-2                                          1/1     Running     0
9h
nautilus-85754d87d7-756qb                       1/1     Running     0
9h
nautilus-85754d87d7-q8j7d                       1/1     Running     0
9h
openapi-5f9cc76544-7fnjm                        1/1     Running     0
9h
openapi-5f9cc76544-vzr7b                        1/1     Running     0
9h
packages-5db49f8b5-lrzhd                        1/1     Running     0
9h
polaris-consul-consul-server-0                  1/1     Running     0
9h
polaris-consul-consul-server-1                  1/1     Running     0
9h
polaris-consul-consul-server-2                  1/1     Running     0
9h
polaris-keycloak-0                              1/1     Running     2
(9h ago)   9h
polaris-keycloak-1                              1/1     Running     0
9h
polaris-keycloak-2                              1/1     Running     0
9h
polaris-keycloak-db-0                           1/1     Running     0
9h
polaris-keycloak-db-1                           1/1     Running     0
9h
polaris-keycloak-db-2                           1/1     Running     0
9h
polaris-mongodb-0                               1/1     Running     0
9h
polaris-mongodb-1                               1/1     Running     0
9h
polaris-mongodb-2                               1/1     Running     0
9h
polaris-ui-66fb99479-qp9gq                      1/1     Running     0
9h
polaris-vault-0                                 1/1     Running     0
9h
polaris-vault-1                                 1/1     Running     0
9h
```

```
polaris-vault-2                                1/1     Running     0
9h
public-metrics-76fbf9594d-zmxzw                1/1     Running     0
9h
storage-backend-metrics-7d7fbc9cb9-lmd25       1/1     Running     0
9h
storage-provider-5bdd456c4b-2fftc              1/1     Running     0
9h
task-service-87575df85-dnn2q                   1/1     Running     3
(9h ago)   9h
task-service-task-purge-28011720--1-q6w4r      0/1     Completed   0
28m
task-service-task-purge-28011735--1-vk6pd      1/1     Running     0
13m
telegraf-ds-2r2kw                              1/1     Running     0
9h
telegraf-ds-6s9d5                              1/1     Running     0
9h
telegraf-ds-96jl7                              1/1     Running     0
9h
telegraf-ds-hbp84                              1/1     Running     0
9h
telegraf-ds-plwzv                              1/1     Running     0
9h
telegraf-ds-sr22c                              1/1     Running     0
9h
telegraf-rs-4sbg8                              1/1     Running     0
9h
telemetry-service-fb9559f7b-mk9l7              1/1     Running     3
(9h ago)   9h
tenancy-559bbc6b48-5msgg                       1/1     Running     0
9h
traefik-d997b8877-7xpf4                        1/1     Running     0
9h
traefik-d997b8877-9xv96                        1/1     Running     0
9h
trident-svc-585c97548c-d25z5                   1/1     Running     0
9h
vault-controller-88484b454-2d6sr               1/1     Running     0
9h
vault-controller-88484b454-fc5cz               1/1     Running     0
9h
vault-controller-88484b454-jktld               1/1     Running     0
9h
```

4. (Optional) Sehen Sie sich den an `acc-operator` Protokolle zur Überwachung des Fortschritts:

```
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.

5. 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  23.10.0-68
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 Einrichtungsverfahren bieten Beispielschritte für einige gängige Typen von Ingress-Controllern.

Bevor Sie beginnen

- 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` Im `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.

Optionen

- 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
```

- So überprüfen Sie die Ausgabe des Astra Control Center CR:

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

Wie es weiter geht

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

Konfigurieren Sie einen externen Zertifikaten-Manager

Wenn bereits ein Cert Manager in Ihrem Kubernetes Cluster vorhanden ist, müssen Sie einige erforderliche Schritte durchführen, damit Astra Control Center keinen eigenen Cert Manager installiert.

Schritte

1. Vergewissern Sie sich, dass ein Zertifikaten-Manager installiert ist:

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

Beispielantwort:

```
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. Erstellen Sie ein Zertifikat-/Schlüsselpaar für das `astraAddress` FQDN:

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

Beispielantwort:

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

3. Erstellen eines Geheimnisses mit zuvor generierten Dateien:

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

Beispielantwort:

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

4. Erstellen Sie ein `ClusterIssuer` Datei, die **genau** die folgenden ist, aber den Namespace-Speicherort enthält, wo Ihr `cert-manager` Pods sind installiert:

```
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
```

Beispielantwort:

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

5. Überprüfen Sie das `ClusterIssuer` Ist richtig aufgekommen. `Ready` Muss sein `True` Bevor Sie fortfahren können:

```
kubectl get ClusterIssuer
```

Beispielantwort:

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

6. Füllen Sie die aus "Astra Control Center-Installationsprozess". Es gibt ein "Erforderlicher Konfigurationsschritt für den Astra Control Center-Cluster YAML" In dem Sie den CRD-Wert ändern, um anzuzeigen, dass der Zertifikaten-Manager extern installiert ist. Sie müssen diesen Schritt während der Installation abschließen, damit das Astra Control Center den externen Zertifikaten-Manager erkennt.

Installieren Sie Astra Control Center mit OpenShift OperatorHub

Wenn Sie Red hat OpenShift verwenden, können Sie Astra Control Center mithilfe des von Red hat zertifizierten Betreibers installieren. Gehen Sie folgendermaßen vor, um Astra Control Center von der zu installieren "Red Hat Ecosystem Catalog" Oder die Red hat OpenShift-Container-Plattform verwenden.

Nach Abschluss dieses Verfahrens müssen Sie zum Installationsvorgang zurückkehren, um den abzuschließen "Verbleibende Schritte" Um die erfolgreiche Installation zu überprüfen, und melden Sie sich an.

Bevor Sie beginnen

- **Umweltvoraussetzungen erfüllen**: "Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor".
- **Gewährleistung einer gesunden Clusteroperatoren und API-Dienste**:
  - Stellen Sie in Ihrem OpenShift-Cluster sicher, dass sich alle Clusterbetreiber in einem ordnungsgemäßen Zustand befinden:

    ```
    oc get clusteroperators
    ```

  - Stellen Sie in Ihrem OpenShift-Cluster sicher, dass sich alle API-Services in einem ordnungsgemäßen Zustand befinden:

    ```
    oc get apiservices
    ```

- **Stellen Sie einen routingfähigen FQDN sicher**: Der Astra FQDN, den Sie verwenden möchten, kann zum Cluster weitergeleitet werden. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.
- **OpenShift-Berechtigungen** erhalten: Sie benötigen alle erforderlichen Berechtigungen und Zugriff auf die Red hat OpenShift Container Platform, um die beschriebenen Installationsschritte auszuführen.
- **Configure a cert Manager**: Wenn ein cert Manager bereits im Cluster vorhanden ist, müssen Sie einige durchführen "Erforderliche Schritte" Damit Astra Control Center nicht seinen eigenen Cert-Manager installiert. Standardmäßig installiert Astra Control Center während der Installation einen eigenen Cert-Manager.
- **Betrachten Sie ein Service-Mesh**: Es wird dringend empfohlen, die Kommunikationskanäle des Astra Control-Host-Clusters mit einem zu sichern "Unterstütztes Service-Mesh".

Istio Service Mesh-Details

Für die Nutzung von Istio Service Mesh müssen Sie Folgendes tun:

- Fügen Sie ein hinzu `istio-injection:enabled` Kennzeichnen Sie den Astra-Namespace, bevor Sie Astra Control Center implementieren.
- Verwenden Sie die `Generic` Einstellung für Eindringen Und bieten eine alternative Ingress für "Externe Lastverteilung".
- Für Red hat OpenShift-Cluster müssen Sie definieren `NetworkAttachmentDefinition` In allen zugehörigen Astra Control Center-Namespaces (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` Für Anwendungscluster oder alle benutzerdefinierten Namespaces, die ersetzt wurden).

```
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
```

- **Kubernetes Ingress-Controller**: Wenn Sie über einen Kubernetes Ingress-Controller verfügen, der externen Zugriff auf Services wie etwa den Lastausgleich in einem Cluster managt, müssen Sie ihn zur Verwendung mit Astra Control Center einrichten:

a. Erstellen Sie den Operator-Namespace:

```
oc create namespace netapp-acc-operator
```

b. "Einrichtung abschließen" Für Ihren Ingress-Controller-Typ.

- **Nur ONTAP-SAN-Treiber**: Wenn Sie einen ONTAP-SAN-Treiber verwenden, stellen Sie sicher, dass Multipath auf allen Kubernetes-Clustern aktiviert ist.

Schritte

- 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
- Suchen Sie die Installationsseite des Bedieners
- Installieren Sie den Operator
- Installieren Sie Astra Control Center

Laden Sie das Astra Control Center herunter und extrahieren Sie es

Sie können das Bundle von Astra Control Center von der NetApp Support-Website herunterladen oder das Bundle mithilfe von Docker aus der Image-Registrierung des Astra Control Service abrufen.

NetApp Support Website

1. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`) Vom "Download-Seite für Astra Control Center".
2. (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 Bündels zu überprüfen.

   Erweitern Sie, um Details anzuzeigen

   ```
   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.

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

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

Astra Control-Image-Registrierung

1. Melden Sie sich beim Astra Control Service an.
2. Wählen Sie im Dashboard **Deploy a self-Managed Instance of Astra Control** aus.
3. Folgen Sie den Anweisungen, um sich bei der Astra Control-Image-Registrierung anzumelden, das Astra Control Center-Installationsabbild zu ziehen und das Image zu extrahieren.

Installieren Sie das NetApp Astra kubectl Plug-in

Sie können das NetApp Astra kubectl Befehlszeilenschnittstelle-Plug-in verwenden, um Images in ein lokales Docker Repository zu verschieben.

Bevor Sie beginnen

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 den sehen `acc.manifest.bundle.yaml` Datei und diese Verzeichnisse:

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

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

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=23.10.0-68
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=23.10.0-68
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://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/23.10.0-68/image:version
```

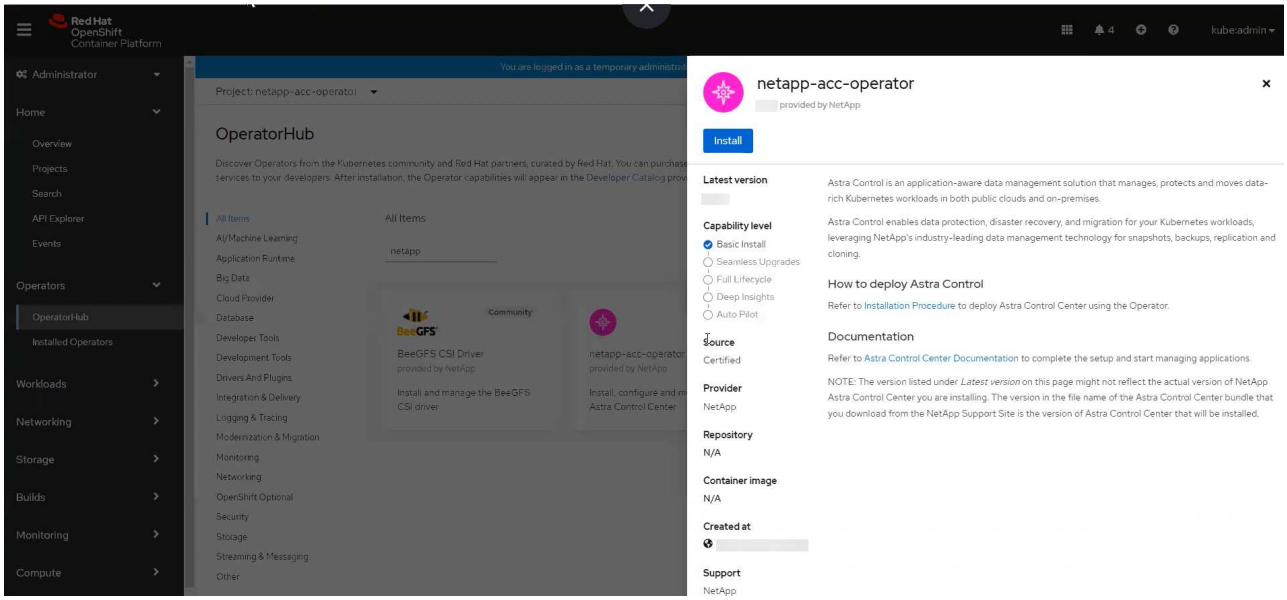
Suchen Sie die Installationsseite des Bedieners

1. Führen Sie eines der folgenden Verfahren aus, um auf die Installationsseite des Bedieners zuzugreifen:

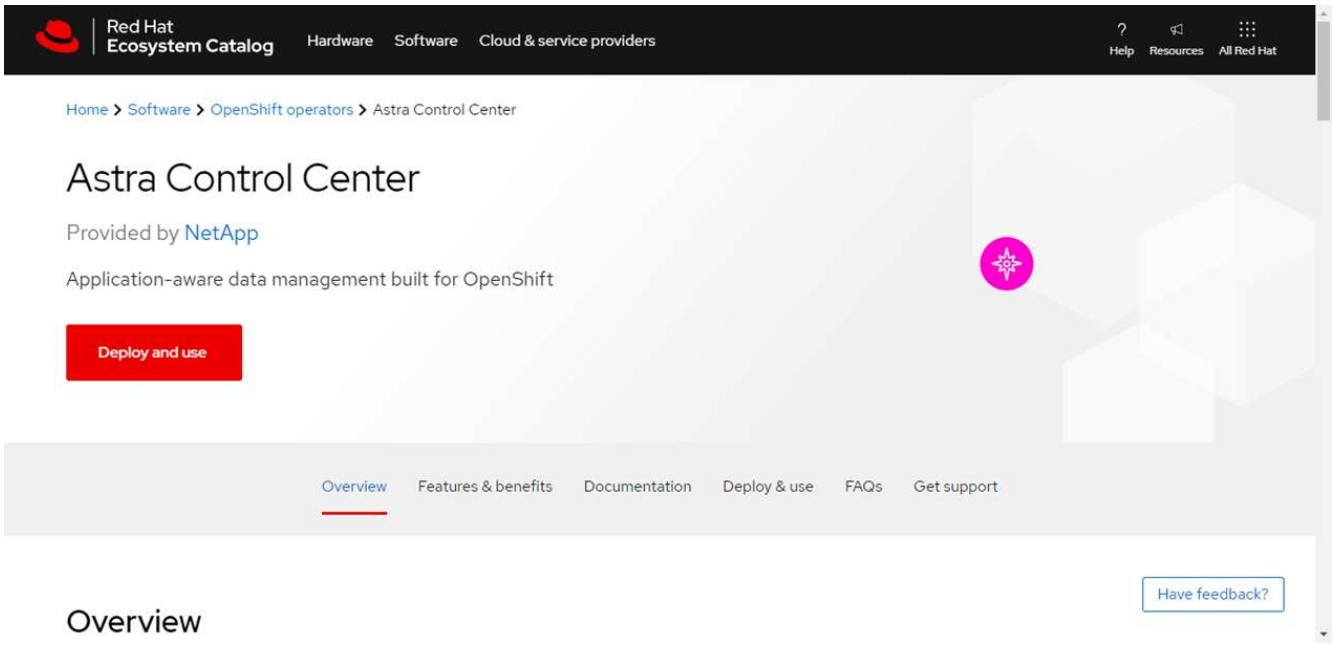
- Über die Red hat OpenShift-Webkonsole:
  i. Melden Sie sich in der OpenShift Container Platform UI an.
  ii. Wählen Sie im Seitenmenü die Option **Operatoren > OperatorHub** aus.

> Mit diesem Operator können Sie nur auf die aktuelle Version von Astra Control Center aktualisieren.

  iii. Suchen Sie nach und wählen Sie den Operator des NetApp Astra Control Center aus.



- Aus Dem Red Hat Ecosystem Catalog:
  i. Wählen Sie das NetApp Astra Control Center aus "Operator".
  ii. Wählen Sie **Bereitstellen und Verwenden**.



Installieren Sie den Operator

1. Füllen Sie die Seite **Install Operator** aus, und installieren Sie den Operator:

   Der Operator ist in allen Cluster-Namespaces verfügbar.

   a. Wählen Sie den Operator-Namespace oder aus `netapp-acc-operator` Der Namespace wird automatisch im Rahmen der Bedienerinstallation erstellt.

   b. Wählen Sie eine manuelle oder automatische Genehmigungsstrategie aus.

      Eine manuelle Genehmigung wird empfohlen. Sie sollten nur eine einzelne Operatorinstanz pro Cluster ausführen.

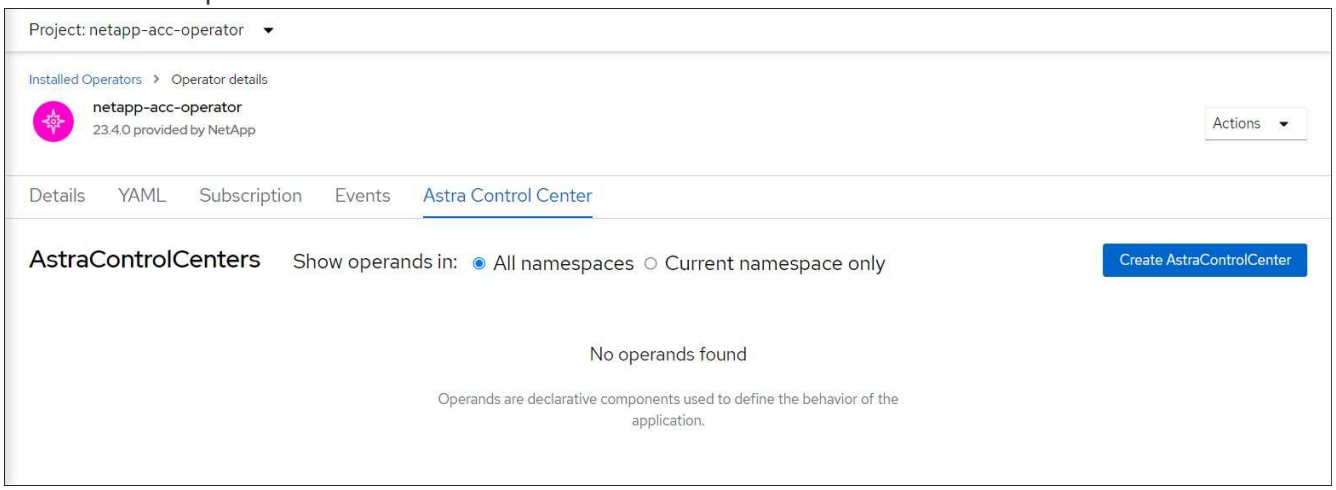
   c. Wählen Sie **Installieren**.

      Wenn Sie eine manuelle Genehmigungsstrategie ausgewählt haben, werden Sie aufgefordert, den manuellen Installationsplan für diesen Operator zu genehmigen.

2. Gehen Sie von der Konsole aus zum OperatorHub-Menü und bestätigen Sie, dass der Operator erfolgreich installiert wurde.

Installieren Sie Astra Control Center

1. Wählen Sie in der Konsole auf der Registerkarte **Astra Control Center** des Astra Control Center-Bedieners die Option **AstraControlCenter erstellen** aus.



2. Füllen Sie die aus `Create AstraControlCenter` Formularfeld:

   a. Behalten Sie den Namen des Astra Control Center bei oder passen Sie diesen an.

   b. Fügen Sie Etiketten für das Astra Control Center hinzu.

   c. Aktivieren oder deaktivieren Sie Auto Support. Es wird empfohlen, die Auto Support-Funktion beizubehalten.

   d. Geben Sie den FQDN des Astra Control Centers oder die IP-Adresse ein. Kommen Sie nicht herein `http://` Oder `https://` Im Adressfeld.

   e. Geben Sie die Astra Control Center-Version ein, z. B. 23.10.0-68.

   f. Geben Sie einen Kontonamen, eine E-Mail-Adresse und einen Administratornamen ein.

g. Wählen Sie eine Richtlinie zur Rückgewinnung von Volumes aus `Retain`, `Recycle`, Oder `Delete`. Der Standardwert ist `Retain`.

h. Wählen Sie die ScaleSize der Installation aus.

> Astra verwendet standardmäßig High Availability (HA). `scaleSize` Von `Medium`, Die die meisten Dienste in HA bereitstellt und mehrere Replikate für Redundanz bereitstellt. Mit `scaleSize` Als `Small`, Astra wird die Anzahl der Replikate für alle Dienste reduzieren, außer für wesentliche Dienste, um den Verbrauch zu reduzieren.

i. Wählen Sie den Typ der Eindringen aus:

- **`Generic`** (`ingressType: "Generic"`) (Standard)

  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" Um Astra Control Center mit einer URL zu zeigen.

- **`AccTraefik`** (`ingressType: "AccTraefik"`)

  Verwenden Sie diese Option, wenn Sie keinen Ingress-Controller konfigurieren möchten. Dies implementiert das Astra Control Center `traefik` Gateway als Service vom Typ 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 Konfiguration des internen DNS-Servers sollten Sie den ausgewählten DNS-Namen für Astra Control Center auf die Load-Balanced IP-Adresse verweisen.

> Weitere Informationen zum Servicetyp „loadbalancer“ und „ingress“ finden Sie unter "Anforderungen".

a. Geben Sie in **Image Registry** Ihren lokalen Container Image Registry-Pfad ein. Kommen Sie nicht herein `http://` Oder `https://` Im Adressfeld.

b. Wenn Sie eine Bildregistrierung verwenden, die eine Authentifizierung erfordert, geben Sie das Bildgeheimnis ein.

> Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, Erstellen Sie ein Geheimnis auf dem Cluster.

c. Geben Sie den Vornamen des Administrators ein.

d. Konfiguration der Ressourcenskalierung

e. Stellen Sie die Standard-Storage-Klasse bereit.

> Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass diese die einzige Storage-Klasse mit der Standardbeschriftung ist.

f. Definieren Sie die Einstellungen für die Verarbeitung von CRD.

3. Wählen Sie die YAML-Ansicht aus, um die ausgewählten Einstellungen zu überprüfen.
4. Wählen Sie `Create`.

Erstellen Sie einen Registrierungsschlüssel

Wenn Sie eine Registrierung verwenden, die eine Authentifizierung erfordert, erstellen Sie einen Schlüssel im OpenShift-Cluster und geben Sie den geheimen Namen in das ein `Create AstraControlCenter` Formularfeld.

1. Erstellen Sie einen Namespace für den Astra Control Center-Betreiber:

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. Erstellen eines Geheimnisses in diesem Namespace:

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

Astra Control unterstützt nur die Geheimnisse der Docker-Registrierung.

3. Füllen Sie die übrigen Felder in aus Das Feld AstraControlCenter-Formular erstellen.

Wie es weiter geht

Füllen Sie die aus "Verbleibende Schritte" Um zu überprüfen, ob Astra Control Center erfolgreich installiert wurde, richten Sie einen Ingress-Controller ein (optional), und melden Sie sich an der UI an. Zusätzlich müssen Sie durchführen "Setup-Aufgaben" Nach Abschluss der Installation.

Installieren Sie Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend

Mit Astra Control Center können Sie Ihre Applikationen in einer Hybrid-Cloud-Umgebung mit automatisierten Kubernetes-Clustern und Cloud Volumes ONTAP Instanzen managen. Astra Control Center kann auch in lokalen Kubernetes-Clustern oder in einem der selbst gemanagten Kubernetes-Cluster in der Cloud-Umgebung implementiert werden.

Mit einer dieser Implementierungen können Sie Applikationsdatenmanagement-Vorgänge mithilfe von Cloud Volumes ONTAP als Storage-Backend durchführen. Außerdem können Sie einen S3-Bucket als Backup-Ziel konfigurieren.

Zur Installation von Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) und Microsoft Azure mit einem Cloud Volumes ONTAP Storage-Backend führen Sie je nach Cloud-Umgebung die folgenden Schritte aus.

- Implementieren Sie Astra Control Center in Amazon Web Services

- Implementieren Sie Astra Control Center in der Google Cloud Platform
- Implementieren Sie Astra Control Center in Microsoft Azure

Applikationen lassen sich in Distributionen mit selbst gemanagten Kubernetes-Clustern managen, wie z. B. mit OpenShift Container Platform (OCP). Nur selbst gemanagte OCP Cluster sind für die Implementierung des Astra Control Center validiert.

Implementieren Sie Astra Control Center in Amazon Web Services

Astra Control Center lässt sich in einem selbst gemanagten Kubernetes-Cluster implementieren, der in einer Public Cloud von Amazon Web Services (AWS) gehostet wird.

Was Sie für AWS benötigen

Vor der Implementierung von Astra Control Center in AWS sind folgende Fragen zu beachten:

- Astra Control Center-Lizenz: Siehe "Lizenzierungsanforderungen für Astra Control Center".
- "Sie erfüllen die Anforderungen des Astra Control Centers".
- NetApp Cloud Central Konto
- Bei Verwendung von OCP, Berechtigungen für die Red hat OpenShift Container Platform (OCP) (auf Namespace-Ebene zum Erstellen von Pods)
- AWS Zugangsdaten, Zugriffs-ID und geheimer Schlüssel mit Berechtigungen, mit denen Sie Buckets und Konnektoren erstellen können
- Zugriff und Anmeldung auf und bei dem AWS Konto Elastic Container Registry (ECR
- Für den Zugriff auf die Astra Control UI sind die Einträge für die gehostete AWS Zone und Amazon Route 53 erforderlich

Anforderungen der Betriebsumgebung für AWS

Astra Control Center erfordert die folgende Betriebsumgebung für AWS:

- Red hat OpenShift Container Platform 4.11 bis 4.13

> Stellen Sie sicher, dass die Betriebsumgebung, die Sie als Host für das Astra Control Center auswählen, den grundlegenden Ressourcenanforderungen in der offiziellen Dokumentation der Umgebung entspricht.

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung die folgenden Ressourcen:

Komponente	Anforderungen
Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität	Mindestens 300 GB verfügbar
Worker-Nodes (AWS EC2 Anforderung)	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM

Komponente	Anforderungen
Load Balancer	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
FQDN	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP, ehemals Cloud Manager)	Astra Trident 23.01 oder höher installiert und konfiguriert und NetApp ONTAP Version 9.9.1 oder neuer als Storage-Backend
Bildregistrierung	NetApp stellt eine Registrierung bereit, mit der Sie Astra Control Center Build-Images abrufen können: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Wenden Sie sich an den NetApp-Support, um Anweisungen zur Verwendung dieser Image-Registrierung während der Installation von Astra Control Center zu erhalten. Wenn Sie nicht auf die NetApp-Image-Registrierung zugreifen können, benötigen Sie eine bestehende private Registrierung, wie z. B. die AWS Elastic Container Registry (ECR), auf die Sie die Build-Images von Astra Control Center übertragen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen. Der gehostete Astra Control Center-Cluster und der verwaltete Cluster müssen Zugriff auf dieselbe Image-Registry haben, um Anwendungen mit dem Restic-basierten Image sichern und wiederherstellen zu können.
Konfiguration von Astra Trident/ONTAP	Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden Kubernetes-Storage-Klassen von ONTAP, die beim Importieren des Kubernetes Clusters in NetApp BlueXP (ehemals Cloud Manager) erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Bei diesen Anforderungen wird davon ausgegangen, dass Astra Control Center die einzige Applikation ist, die in der Betriebsumgebung ausgeführt wird. Wenn in der Umgebung zusätzliche Applikationen ausgeführt werden, passen Sie diese Mindestanforderungen entsprechend an.

Das AWS-Registry-Token läuft innerhalb von 12 Stunden ab. Danach müssen Sie das Secret der Docker-Image-Registrierung verlängern.

Überblick über die Implementierung für AWS

Hier finden Sie eine Übersicht über die Vorgehensweise zur Installation des Astra Control Center für AWS mit Cloud Volumes ONTAP als Storage-Backend.

Jeder dieser Schritte wird unten im Detail erklärt.

1. dass Sie über ausreichende IAM-Berechtigungen verfügen.
2. Installation eines RedHat OpenShift-Clusters in AWS.
3. Konfigurieren von AWS.
4. Konfiguration von NetApp BlueXP für AWS.
5. Installieren Sie Astra Control Center für AWS.

Stellen Sie sicher, dass Sie über ausreichende IAM-Berechtigungen verfügen

Stellen Sie sicher, dass Sie über ausreichende IAM-Rollen und -Berechtigungen verfügen, mit denen Sie ein RedHat OpenShift Cluster und einen NetApp BlueXP (ehemals Cloud Manager) Connector installieren können.

Siehe "Erste AWS Zugangsdaten".

Installation eines RedHat OpenShift-Clusters in AWS

Installation eines RedHat OpenShift-Container-Plattform-Clusters auf AWS

Installationsanweisungen finden Sie unter "Installation eines Clusters auf AWS in OpenShift Container Platform".

Konfigurieren von AWS

Konfigurieren Sie als nächstes AWS, um ein virtuelles Netzwerk zu erstellen, EC2 Computing-Instanzen einzurichten und einen AWS S3-Bucket zu erstellen. Wenn Sie nicht auf den zugreifen können NetApp Astra Control Center Image-RegistrierungSie müssen auch eine Elastic Container Registry (ECR) erstellen, um die Astra Control Center-Images zu hosten und die Bilder in diese Registry zu verschieben.

Folgen Sie der AWS Dokumentation, um die folgenden Schritte auszuführen. Siehe "AWS Installationsdokumentation".

1. Virtuelles AWS Netzwerk erstellen.
2. EC2 Computing-Instanzen prüfen. Dabei können es sich um einen Bare Metal Server oder VMs in AWS handelt.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in AWS, um die Astra-Anforderungen zu erfüllen. Siehe "Anforderungen des Astra Control Centers".

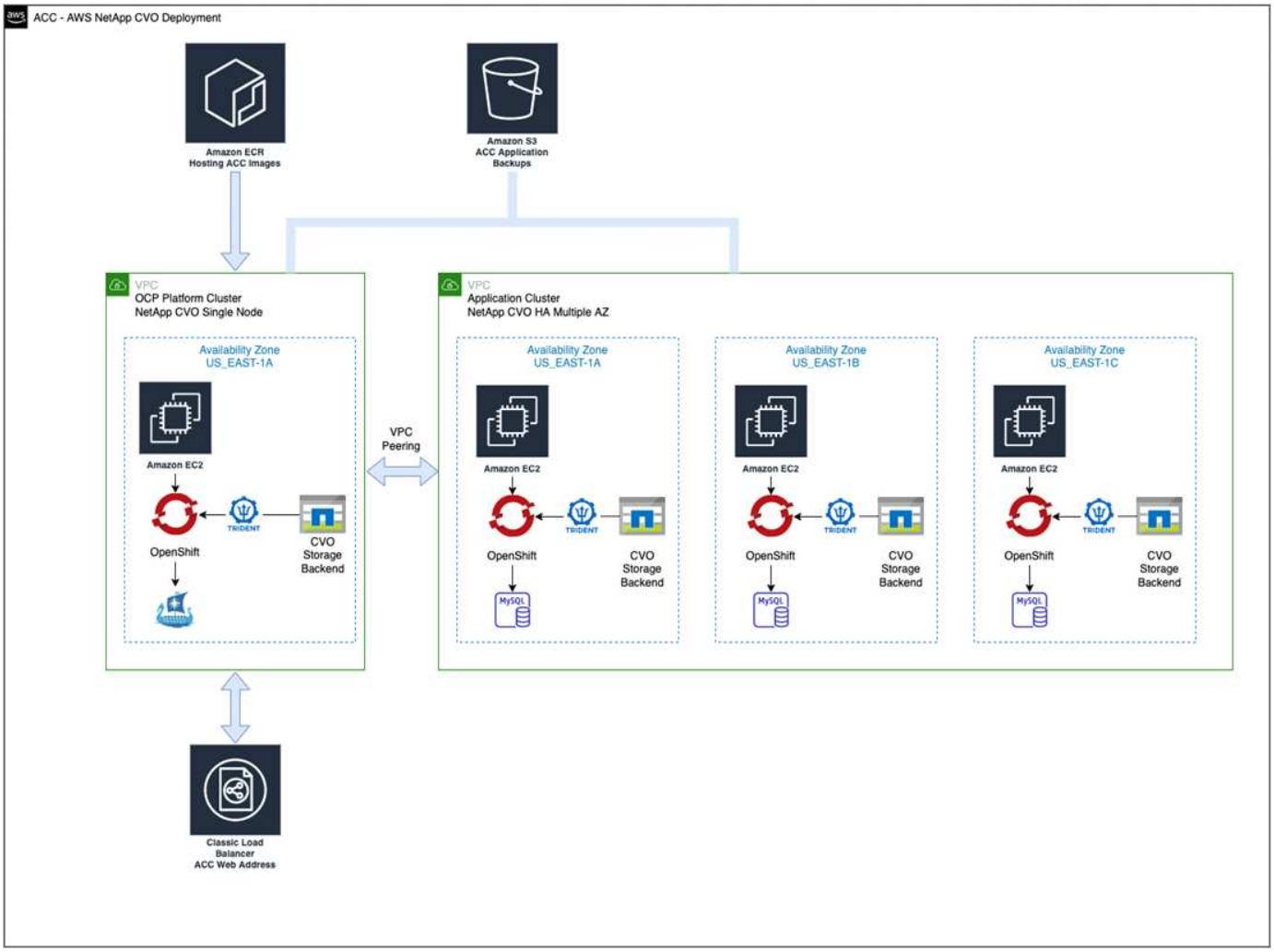
4. Erstellen Sie mindestens einen AWS S3-Bucket zum Speichern Ihrer Backups.
5. (Optional) Wenn Sie nicht auf den zugreifen können NetApp-Image-Registrierung, Gehen Sie wie folgt vor:
   a. Eine AWS Elastic Container Registry (ECR) erstellen, um alle Astra Control Center Images zu hosten.

> Wenn Sie den ECR nicht erstellen, kann Astra Control Center mit einem AWS Backend nicht auf die Monitoring-Daten von einem Cluster mit Cloud Volumes ONTAP zugreifen. Das Problem wird verursacht, wenn der Cluster, den Sie mit Astra Control Center ermitteln und verwalten möchten, keinen AWS ECR-Zugriff hat.

   b. Übertragen Sie die Astra Control Center Images in Ihre definierte Registrierung.

> Das AWS Elastic Container Registry (ECR) Token läuft nach 12 Stunden ab und verursacht das Fehlschlagen clusterübergreifender Klonvorgänge. Dieses Problem tritt auf, wenn ein Storage-Back-End von für AWS konfigurierten Cloud Volumes ONTAP gemanagt wird. Um dieses Problem zu beheben, müssen Sie sich erneut mit der ECR authentifizieren und ein neues Geheimnis generieren, damit Klonvorgänge erfolgreich fortgesetzt werden können.

Beispiel für eine AWS Implementierung:



Konfiguration von NetApp BlueXP für AWS

Erstellen Sie mit NetApp BlueXP (früher Cloud Manager) einen Workspace, fügen Sie eine Connector zu AWS hinzu, erstellen Sie eine Arbeitsumgebung und importieren Sie das Cluster.

Befolgen Sie die BlueXP-Dokumentation, um die folgenden Schritte auszuführen. Siehe folgendes:

- "Erste Schritte mit Cloud Volumes ONTAP in AWS".
- "Erstellen Sie einen Connector in AWS mit BlueXP"

Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu.
2. Erstellen Sie einen Arbeitsbereich.
3. Fügen Sie einen Connector für AWS hinzu. Entscheiden Sie sich für AWS als Provider.
4. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
   a. Ort: „Amazon Web Services (AWS)“
   b. Typ: „Cloud Volumes ONTAP HA“
5. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.
   a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s** > **Cluster list** > **Cluster-Details** wählen.
   b. Beachten Sie in der oberen rechten Ecke die Astra Trident-Version.
   c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen, für die NetApp als provisionierung angezeigt wird.

      Dies importiert Ihr Red hat OpenShift-Cluster und weist ihm eine Standardspeicherklasse zu. Sie wählen die Speicherklasse aus.
      Astra Trident wird automatisch im Rahmen des Imports und der Erkennung installiert.
6. Beachten Sie alle persistenten Volumes und Volumes in dieser Cloud Volumes ONTAP-Implementierung.

> Cloud Volumes ONTAP kann als Single Node oder in High Availability betrieben werden. Wenn HA aktiviert ist, notieren Sie den HA-Status und den Implementierungsstatus der Nodes, die in AWS ausgeführt werden.

Installieren Sie Astra Control Center für AWS

Dem Standard folgen "Installationsanweisungen für Astra Control Center".

> AWS verwendet den Bucket-Typ generischer S3.

Implementieren Sie Astra Control Center in der Google Cloud Platform

Astra Control Center lässt sich in einem selbst gemanagten Kubernetes-Cluster implementieren, der auf einer Google Cloud Platform (GCP) Public Cloud gehostet wird.

Was wird für GCP benötigt

Vor der Implementierung von Astra Control Center in GCP sind folgende Elemente erforderlich:

- Astra Control Center-Lizenz: Siehe "Lizenzierungsanforderungen für Astra Control Center".
- "Sie erfüllen die Anforderungen des Astra Control Centers".
- NetApp Cloud Central Konto
- Bei Verwendung von OCP, Red hat OpenShift Container Platform (OCP) 4.11 bis 4.13
- Bei Verwendung von OCP, Berechtigungen für die Red hat OpenShift Container Platform (OCP) (auf Namespace-Ebene zum Erstellen von Pods)
- GCP-Servicekonto mit Berechtigungen, mit denen Sie Buckets und Konnektoren erstellen können

Anforderungen an die Betriebsumgebung für GCP

Stellen Sie sicher, dass die Betriebsumgebung, die Sie als Host für das Astra Control Center auswählen, den grundlegenden Ressourcenanforderungen in der offiziellen Dokumentation der Umgebung entspricht.

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung die folgenden Ressourcen:

Komponente	Anforderungen
Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität	Mindestens 300 GB verfügbar
Worker-Nodes (GCP-Compute-Anforderung)	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
Load Balancer	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
FQDN (GCP-DNS-ZONE)	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP, ehemals Cloud Manager)	Astra Trident 23.01 oder höher installiert und konfiguriert und NetApp ONTAP Version 9.9.1 oder höher als Storage-Backend

Komponente	Anforderungen
Bildregistrierung	NetApp stellt eine Registrierung bereit, mit der Sie Astra Control Center Build-Images abrufen können: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Wenden Sie sich an den NetApp-Support, um Anweisungen zur Verwendung dieser Image-Registrierung während der Installation von Astra Control Center zu erhalten. Wenn Sie nicht auf die NetApp-Image-Registrierung zugreifen können, benötigen Sie eine bestehende private Registrierung, wie z. B. die Google-Container-Registrierung, auf die Sie die Build-Images des Astra Control Centers übertragen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen. Sie müssen anonymen Zugriff aktivieren, um Restic Images für Backups zu erstellen.
Konfiguration von Astra Trident/ONTAP	Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden ONTAP Kubernetes Storage-Klassen, die beim Import des Kubernetes Clusters in NetApp BlueXP erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Bei diesen Anforderungen wird davon ausgegangen, dass Astra Control Center die einzige Applikation ist, die in der Betriebsumgebung ausgeführt wird. Wenn in der Umgebung zusätzliche Applikationen ausgeführt werden, passen Sie diese Mindestanforderungen entsprechend an.

Übersicht über die Implementierung für GCP

Hier ist eine Übersicht über die Vorgehensweise bei der Installation des Astra Control Center auf einem selbst verwalteten OCP-Cluster in GCP mit Cloud Volumes ONTAP als Storage-Backend.

Jeder dieser Schritte wird unten im Detail erklärt.

1. Installieren Sie einen RedHat OpenShift-Cluster auf GCP.
2. Erstellung eines GCP-Projekts und einer virtuellen Private Cloud.
3. dass Sie über ausreichende IAM-Berechtigungen verfügen.

4. GCP konfigurieren.
5. NetApp BlueXP für GCP konfigurieren.
6. Astra Control Center für GCP installieren.

Installieren Sie einen RedHat OpenShift-Cluster auf GCP

Der erste Schritt ist die Installation eines RedHat OpenShift-Clusters auf GCP.

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

- "Installation eines OpenShift-Clusters in GCP"
- "Erstellen eines GCP-Service-Kontos"

Erstellung eines GCP-Projekts und einer virtuellen Private Cloud

Erstellung von mindestens einem GCP-Projekt und einer Virtual Private Cloud (VPC).

> OpenShift kann möglicherweise eigene Ressourcengruppen erstellen. Darüber hinaus sollte auch eine GCP VPC definiert werden. Siehe OpenShift-Dokumentation.

Sie können eine Plattformcluster-Ressourcengruppe und eine Zielapplikation OpenShift-Cluster-Ressourcengruppe erstellen.

Stellen Sie sicher, dass Sie über ausreichende IAM-Berechtigungen verfügen

Stellen Sie sicher, dass Sie über ausreichende IAM-Rollen und -Berechtigungen verfügen, mit denen Sie ein RedHat OpenShift Cluster und einen NetApp BlueXP (ehemals Cloud Manager) Connector installieren können.

Siehe "Erste GCP-Zugangsdaten und -Berechtigungen".

GCP konfigurieren

Konfigurieren Sie anschließend GCP für die Erstellung einer VPC, die Einrichtung von Computing-Instanzen und die Erstellung eines Google Cloud Object Storage. Wenn Sie nicht auf den zugreifen können NetApp Astra Control Center Image-Registrierung, Sie müssen auch eine Google Container Registry erstellen, um die Astra Control Center-Bilder zu hosten, und die Bilder auf diese Registrierung zu schieben.

Befolgen Sie die GCP-Dokumentation, um die folgenden Schritte auszuführen. Siehe Installieren des OpenShift-Clusters in GCP.

1. Erstellen eines GCP-Projekts und der VPC in der GCP, die Sie für den OCP-Cluster mit dem CVO-Back-End verwenden möchten
2. Prüfen Sie die Computing-Instanzen. Dabei kann es sich um einen Bare Metal Server oder VMs in GCP handelt.
3. Wenn der Instanztyp nicht bereits den Astra-Mindestanforderungen für die Ressourcen für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in GCP, um die Astra-Anforderungen zu erfüllen. Siehe "Anforderungen des Astra Control Centers".
4. Erstellen Sie mindestens einen GCP Cloud Storage Bucket, um Ihre Backups zu speichern.
5. Erstellen eines Geheimnisses, das für den Bucket-Zugriff erforderlich ist
6. (Optional) Wenn Sie nicht auf den zugreifen können NetApp-Image-Registrierung, Gehen Sie wie folgt vor:

a. Erstellen Sie eine Google Container Registry, um die Astra Control Center-Images zu hosten.
b. Richten Sie Google Container Registry-Zugriff für Docker Push/Pull für alle Astra Control Center-Bilder ein.

   Beispiel: Astra Control Center-Bilder können in diese Registrierung verschoben werden, indem das folgende Skript eingegeben wird:

   ```
   gcloud auth activate-service-account <service account email address>
   --key-file=<GCP Service Account JSON file>
   ```

   Dieses Skript erfordert eine Astra Control Center Manifest-Datei und Ihren Google Image Registry-Speicherort. Beispiel:

```
manifestfile=acc.manifest.bundle.yaml
GCP_CR_REGISTRY=<target GCP image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
    docker push $GCP_CR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

7. Richten Sie DNS-Zonen ein.

NetApp BlueXP für GCP konfigurieren

Erstellen Sie mithilfe von NetApp BlueXP (früher Cloud Manager) einen Workspace, fügen Sie eine Connector zur GCP hinzu, erstellen Sie eine Arbeitsumgebung und importieren Sie das Cluster.

Befolgen Sie die BlueXP-Dokumentation, um die folgenden Schritte auszuführen. Siehe "Erste Schritte mit Cloud Volumes ONTAP in GCP".

Bevor Sie beginnen

- Zugriff auf das GCP-Servicekonto mit den erforderlichen IAM-Berechtigungen und -Rollen

Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu. Siehe "GCP-Konten hinzufügen".
2. Fügen Sie einen Connector für GCP hinzu.
   a. Entscheiden Sie sich für „GCP“ als Provider.
   b. GCP-Zugangsdaten eingeben. Siehe "Erstellen eines Connectors in GCP von BlueXP".
   c. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.

3. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
   a. Ort: „GCP“
   b. Typ: „Cloud Volumes ONTAP HA“
4. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.
   a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s** > **Cluster list** > **Cluster-Details** wählen.
   b. Beachten Sie oben rechts die Trident-Version.
   c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen mit „NetApp“ als provisionierung.

      Dies importiert Ihr Red hat OpenShift-Cluster und weist ihm eine Standardspeicherklasse zu. Sie wählen die Speicherklasse aus.
      Astra Trident wird automatisch im Rahmen des Imports und der Erkennung installiert.
5. Beachten Sie alle persistenten Volumes und Volumes in dieser Cloud Volumes ONTAP-Implementierung.

Cloud Volumes ONTAP kann als Single Node oder in High Availability (HA) betrieben werden. Wenn HA aktiviert ist, notieren Sie den HA-Status und den Node-Implementierungsstatus, der in GCP ausgeführt wird.

Astra Control Center für GCP installieren

Dem Standard folgen "Installationsanweisungen für Astra Control Center".

GCP verwendet den allgemeinen S3-Bucket-Typ.

1. Generieren Sie das Docker Secret, um Bilder für die Astra Control Center-Installation zu übertragen:

```
kubectl create secret docker-registry <secret name> --docker
-server=<Registry location> --docker-username=_json_key --docker
-password="$(cat <GCP Service Account JSON file>)" --namespace=pcloud
```

Implementieren Sie Astra Control Center in Microsoft Azure

Astra Control Center lässt sich in einem selbst gemanagten Kubernetes-Cluster implementieren, der in einer Microsoft Azure Public Cloud gehostet wird.

Was Sie für Azure benötigen

Vor der Implementierung von Astra Control Center in Azure sind folgende Fragen erforderlich:

- Astra Control Center-Lizenz: Siehe "Lizenzierungsanforderungen für Astra Control Center".
- "Sie erfüllen die Anforderungen des Astra Control Centers".
- NetApp Cloud Central Konto
- Bei Verwendung von OCP, Red hat OpenShift Container Platform (OCP) 4.11 bis 4.13
- Bei Verwendung von OCP, Berechtigungen für die Red hat OpenShift Container Platform (OCP) (auf Namespace-Ebene zum Erstellen von Pods)

- Azure Zugangsdaten mit Berechtigungen, mit denen Sie Buckets und Konnektoren erstellen können

Anforderungen an die Betriebsumgebung für Azure

Stellen Sie sicher, dass die Betriebsumgebung, die Sie als Host für das Astra Control Center auswählen, den grundlegenden Ressourcenanforderungen in der offiziellen Dokumentation der Umgebung entspricht.

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung die folgenden Ressourcen:

Siehe "Anforderungen an die Betriebsumgebung des Astra Control Centers".

Komponente	Anforderungen
Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität	Mindestens 300 GB verfügbar
Worker-Nodes (Azure-Computing-Anforderung)	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
Load Balancer	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
FQDN (Azure-DNS-Zone)	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP)	Astra Trident 23.01 oder höher installiert und konfiguriert und NetApp ONTAP Version 9.9.1 oder höher wird als Storage-Backend verwendet
Bildregistrierung	NetApp stellt eine Registrierung bereit, mit der Sie Astra Control Center Build-Images abrufen können: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Wenden Sie sich an den NetApp-Support, um Anweisungen zur Verwendung dieser Image-Registrierung während der Installation von Astra Control Center zu erhalten. Wenn Sie nicht auf die NetApp-Image-Registrierung zugreifen können, benötigen Sie eine bestehende private Registrierung, wie z. B. die Azure-Container-Registrierung (ACR), auf die Sie die Build-Images des Astra Control Centers übertragen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen. Sie müssen anonymen Zugriff aktivieren, um Restic Images für Backups zu erstellen.

Komponente	Anforderungen
Konfiguration von Astra Trident/ONTAP	Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden ONTAP Kubernetes Storage-Klassen, die beim Import des Kubernetes Clusters in NetApp BlueXP erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Bei diesen Anforderungen wird davon ausgegangen, dass Astra Control Center die einzige Applikation ist, die in der Betriebsumgebung ausgeführt wird. Wenn in der Umgebung zusätzliche Applikationen ausgeführt werden, passen Sie diese Mindestanforderungen entsprechend an.

Überblick über die Implementierung für Azure

Hier finden Sie eine Übersicht über die Vorgehensweise zur Installation von Astra Control Center für Azure.

Jeder dieser Schritte wird unten im Detail erklärt.

1. Installieren Sie einen RedHat OpenShift-Cluster auf Azure.
2. Erstellen von Azure Ressourcengruppen.
3. dass Sie über ausreichende IAM-Berechtigungen verfügen.
4. Konfigurieren Sie Azure.
5. Konfiguration von NetApp BlueXP (ehemals Cloud Manager) für Azure.
6. Installation und Konfiguration von Astra Control Center für Azure.

Installieren Sie einen RedHat OpenShift-Cluster auf Azure

Der erste Schritt ist die Installation eines RedHat OpenShift-Clusters unter Azure.

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

- "OpenShift-Cluster wird auf Azure installiert".
- "Installieren eines Azure-Kontos".

Erstellen von Azure Ressourcengruppen

Erstellen Sie mindestens eine Azure-Ressourcengruppe.

OpenShift kann möglicherweise eigene Ressourcengruppen erstellen. Zusätzlich sollten Sie auch Azure-Ressourcengruppen definieren. Siehe OpenShift-Dokumentation.

Sie können eine Plattformcluster-Ressourcengruppe und eine Zielapplikation OpenShift-Cluster-Ressourcengruppe erstellen.

Stellen Sie sicher, dass Sie über ausreichende IAM-Berechtigungen verfügen

Stellen Sie sicher, dass Sie über ausreichende IAM-Rollen und -Berechtigungen verfügen, mit denen Sie ein RedHat OpenShift-Cluster und einen NetApp BlueXP Connector installieren können.

Siehe "Azure Zugangsdaten und Berechtigungen".

Konfigurieren Sie Azure

Konfigurieren Sie als nächstes Azure, um ein virtuelles Netzwerk zu erstellen, Compute-Instanzen einzurichten und einen Azure Blob-Container zu erstellen. Wenn Sie nicht auf den zugreifen können NetApp Astra Control Center Image-RegistrierungSie müssen auch eine Azure Container Registry (ACR) erstellen, um die Astra Control Center-Images zu hosten und die Bilder in diese Registrierung zu verschieben.

Folgen Sie der Azure-Dokumentation, um die folgenden Schritte durchzuführen. Siehe "OpenShift-Cluster wird auf Azure installiert".

1. Virtuelles Azure Netzwerk erstellen.
2. Prüfen Sie die Computing-Instanzen. Dabei können es sich um einen Bare Metal Server oder VMs in Azure handelt.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in Azure, um die Astra-Anforderungen zu erfüllen. Siehe "Anforderungen des Astra Control Centers".
4. Erstellen Sie mindestens einen Azure Blob Container, um Ihre Backups zu speichern.
5. Erstellen Sie ein Speicherkonto. Sie benötigen ein Storage-Konto, um einen Container zu erstellen, der im Astra Control Center als Bucket verwendet wird.
6. Erstellen eines Geheimnisses, das für den Bucket-Zugriff erforderlich ist
7. (Optional) Wenn Sie nicht auf den zugreifen können NetApp-Image-Registrierung, Gehen Sie wie folgt vor:
    a. Azure Container Registry (ACR) erstellen, um die Astra Control Center Images zu hosten.
    b. ACR-Zugriff für Docker Push/Pull für alle Astra Control Center Images einrichten
    c. Übertragen Sie die Astra Control Center-Images mithilfe des folgenden Skripts in diese Registrierung:

```
az acr login -n <AZ ACR URL/Location>
This script requires the Astra Control Center manifest file and your
Azure ACR location.
```

    - Beispiel*:

```
manifestfile=acc.manifest.bundle.yaml
AZ_ACR_REGISTRY=<target Azure ACR image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

8. Richten Sie DNS-Zonen ein.

Konfiguration von NetApp BlueXP (ehemals Cloud Manager) für Azure

Erstellen Sie mit BlueXP (früher Cloud Manager) einen Workspace, fügen Sie einen Connector zu Azure hinzu, erstellen Sie eine Arbeitsumgebung und importieren Sie das Cluster.

Befolgen Sie die BlueXP-Dokumentation, um die folgenden Schritte auszuführen. Siehe "Erste Schritte mit BlueXP in Azure".

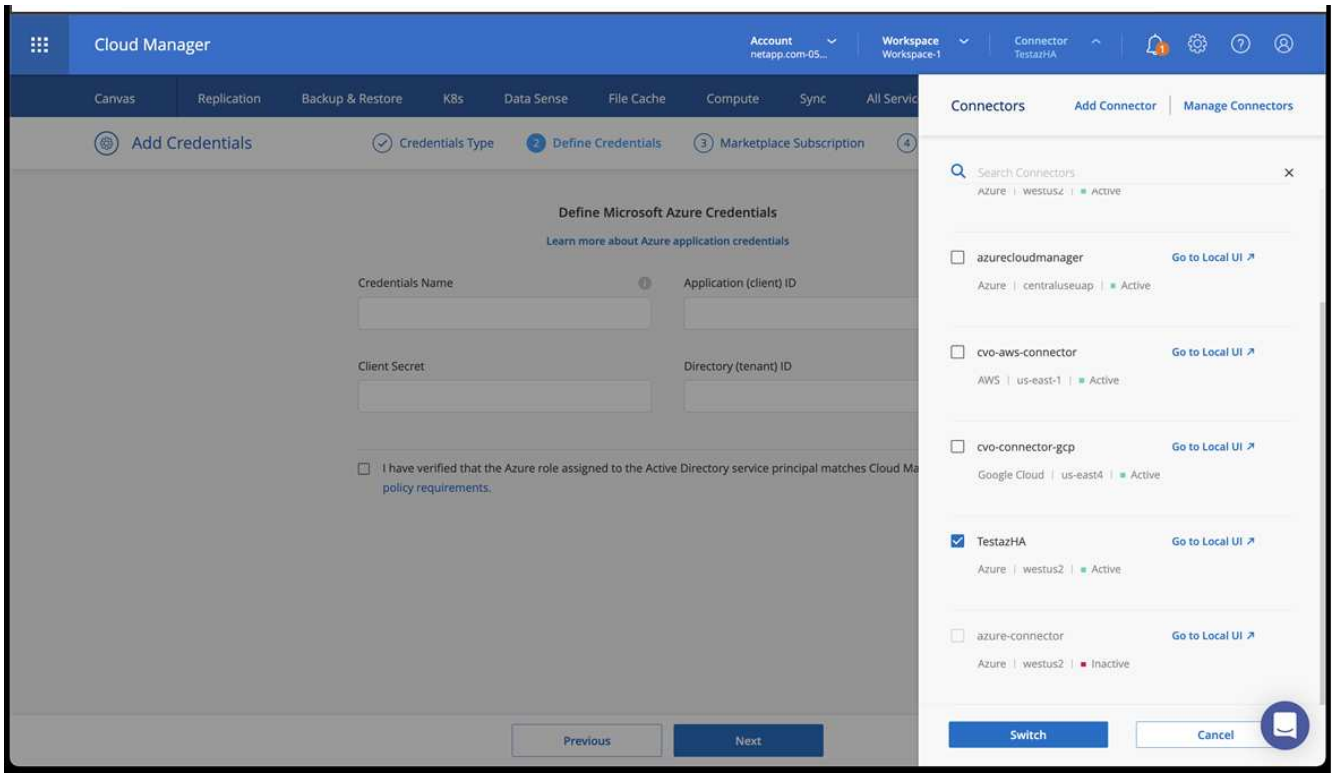
Bevor Sie beginnen

Zugriff auf das Azure Konto mit den erforderlichen IAM-Berechtigungen und -Rollen

Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu.
2. Fügen Sie einen Connector für Azure hinzu. Siehe "BlueXP-Richtlinien".
   a. Wählen Sie als Provider * Azure* aus.
   b. Geben Sie die Azure-Zugangsdaten ein, einschließlich der Anwendungs-ID, des Client-Geheimdienstes und der Verzeichniskennung (Mandanten).

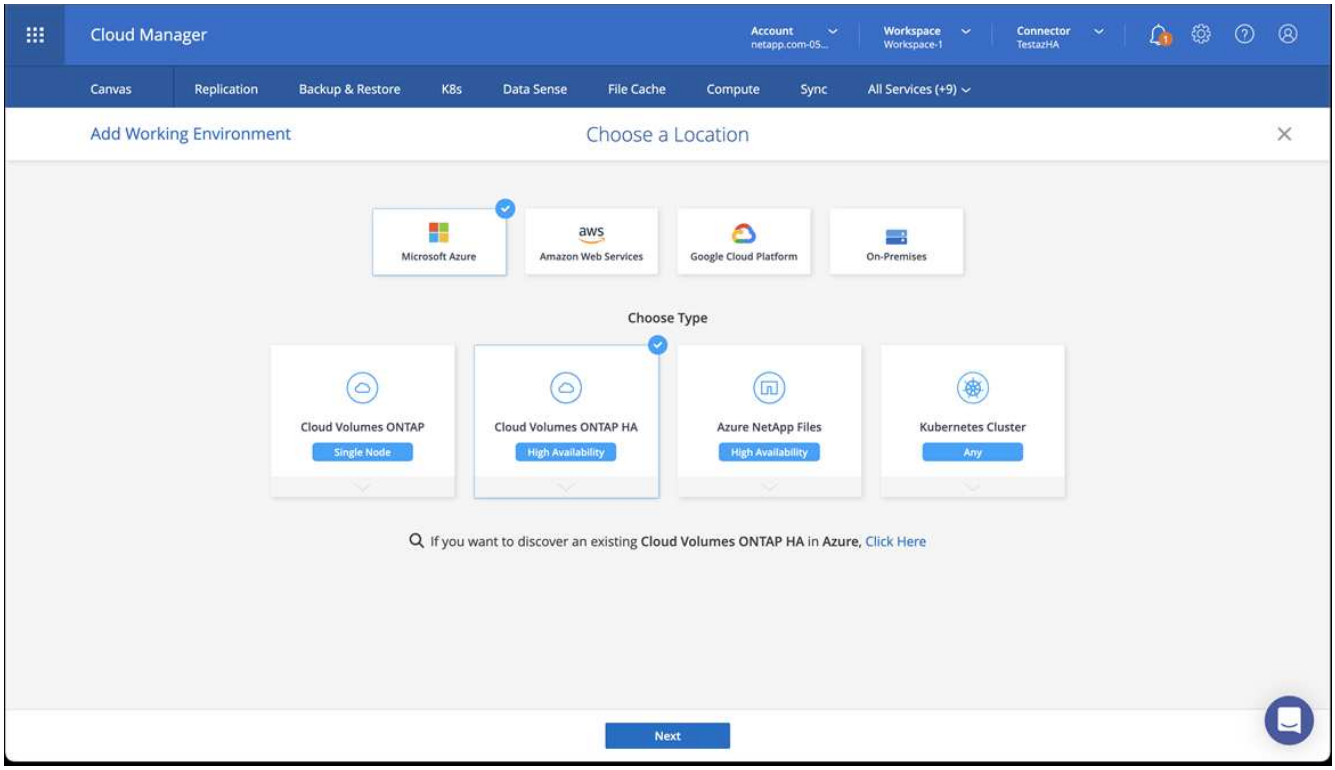
      Siehe "Erstellen eines Konnektors in Azure aus BlueXPr".
3. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.



4. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.

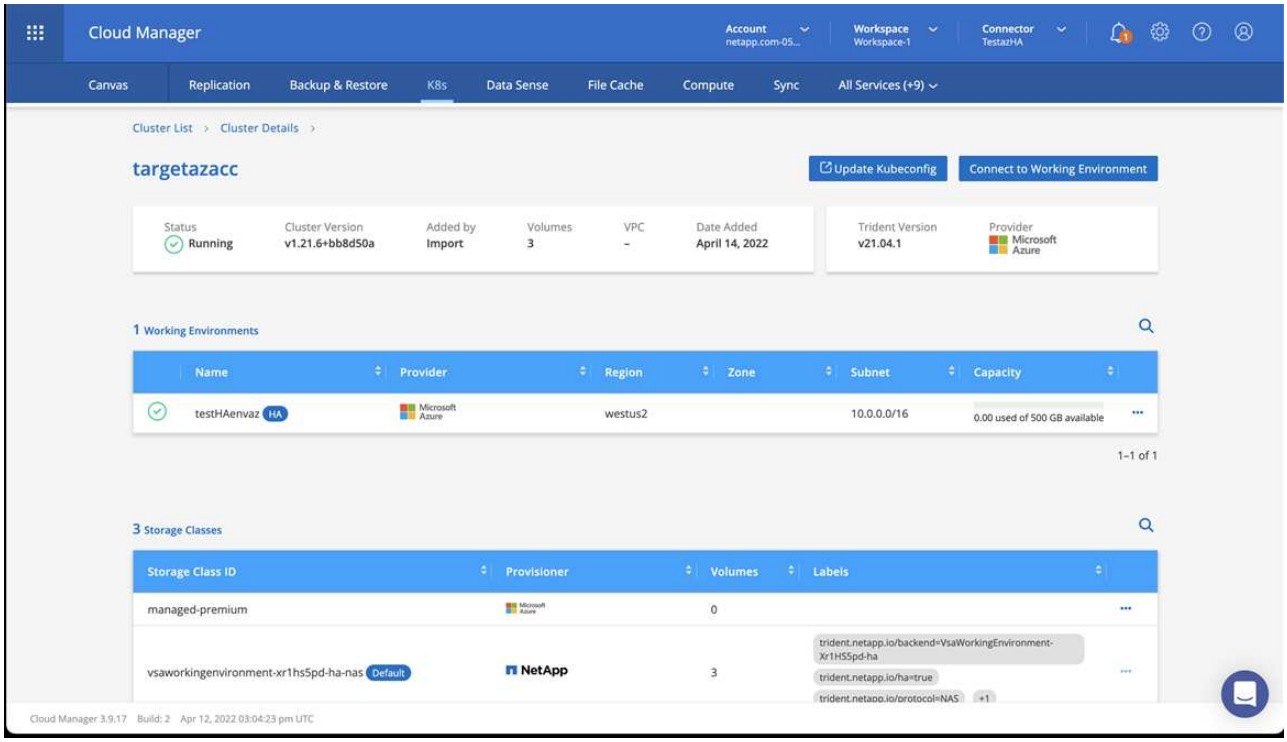
   a. Ort: „Microsoft Azure“.

   b. Typ: „Cloud Volumes ONTAP HA“.



5. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.

a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s** > **Cluster list** > **Cluster-Details** wählen.



b. Beachten Sie in der oberen rechten Ecke die Astra Trident-Version.

c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen, für die NetApp als provisionierung angezeigt wird.

Damit wird Ihr Red hat OpenShift-Cluster importiert und eine Standardspeicherklasse zugewiesen. Sie wählen die Speicherklasse aus.
Astra Trident wird automatisch im Rahmen des Imports und der Erkennung installiert.

6. Beachten Sie alle persistenten Volumes und Volumes in dieser Cloud Volumes ONTAP-Implementierung.
7. Cloud Volumes ONTAP kann als Single Node oder in High Availability betrieben werden. Wenn HA aktiviert ist, notieren Sie den HA-Status und den Node-Implementierungsstatus, der in Azure ausgeführt wird.

Installation und Konfiguration von Astra Control Center für Azure

Installieren Sie Astra Control Center standardmäßig "Installationsanweisungen".

Fügen Sie über Astra Control Center einen Azure-Bucket hinzu. Siehe "Astra Control Center einrichten und Buckets hinzufügen".

Konfigurieren Sie nach der Installation das Astra Control Center

Je nach Umgebung kann es nach der Installation des Astra Control Center zusätzliche Konfigurationsmöglichkeiten geben.

Ressourceneinschränkungen entfernen

In einigen Umgebungen werden die Objekte ResourceQuotas und LimitRanges verwendet, um zu verhindern, dass die Ressourcen in einem Namespace alle verfügbaren CPUs und Speicher im Cluster verbrauchen. Das Astra Control Center stellt keine Höchstgrenzen ein, sodass diese Ressourcen nicht eingehalten werden. Wenn Ihre Umgebung auf diese Weise konfiguriert ist, müssen Sie diese Ressourcen aus den Namespaces entfernen, in denen Sie Astra Control Center installieren möchten.

Sie können folgende Schritte verwenden, um diese Kontingente und Grenzen abzurufen und zu entfernen. In diesen Beispielen wird die Befehlsausgabe direkt nach dem Befehl angezeigt.

Schritte

1. Erhalten Sie die Ressourcen-Kontingente im `netapp-acc` (Oder benutzerdefinierter Name) Namespace:

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

Antwort:

```
NAME          AGE   REQUEST                                              LIMIT
pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
limits.cpu: 0/200, limits.memory: 0/1000Gi
pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
limits.cpu: 0/2, limits.memory: 0/2Gi
pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
limits.cpu: 0/20, limits.memory: 0/200Gi
```

2. Alle Ressourcen-Kontingente nach Namen löschen:

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom
namespace]
```

3. Erhalten Sie die Grenzwerte im `netapp-acc` (Oder benutzerdefinierter Name) Namespace:

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

Antwort:

```
NAME              CREATED AT
cpu-limit-range   2022-06-27T19:01:23Z
```

4. Grenzwerte nach Namen löschen:

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom
namespace]
```

Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

Astra Control Center verwendet standardmäßig ein selbstsigniertes TLS-Zertifikat für Ingress-Controller-Datenverkehr (nur in bestimmten Konfigurationen) und die Web-UI-Authentifizierung mit Webbrowsern. Sie können das vorhandene selbst signierte TLS-Zertifikat entfernen und durch ein TLS-Zertifikat ersetzen, das von einer Zertifizierungsstelle (CA) signiert ist.

Das selbstsignierte Standardzertifikat wird für zwei Verbindungstypen verwendet:

- HTTPS-Verbindungen zur Web-UI des Astra Control Center
- Ingress-Controller-Verkehr (nur wenn der `ingressType: "AccTraefik"` Das Hotel wurde in der eingerichtet `astra_control_center.yaml` Datei während Astra Control Center Installation)

Durch Ersetzen des Standard-TLS-Zertifikats wird das Zertifikat ersetzt, das für die Authentifizierung für diese Verbindungen verwendet wird.

Bevor Sie beginnen

- Kubernetes-Cluster mit installiertem Astra Control Center
- Administratorzugriff auf eine Command Shell auf dem zu ausgeführten Cluster `kubectl` Befehle
- Private Schlüssel- und Zertifikatdateien aus der CA

Entfernen Sie das selbstsignierte Zertifikat

Entfernen Sie das vorhandene selbstsignierte TLS-Zertifikat.

1. Melden Sie sich mit SSH beim Kubernetes Cluster an, der als administrativer Benutzer Astra Control Center hostet.
2. Suchen Sie das mit dem aktuellen Zertifikat verknüpfte TLS-Geheimnis mit dem folgenden Befehl, Ersetzen `<ACC-deployment-namespace>` Mit dem Astra Control Center Deployment Namespace:

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

3. Löschen Sie den derzeit installierten Schlüssel und das Zertifikat mit den folgenden Befehlen:

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

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

Fügen Sie mithilfe der Befehlszeile ein neues Zertifikat hinzu

Fügen Sie ein neues TLS-Zertifikat hinzu, das von einer CA signiert wird.

1. Verwenden Sie den folgenden Befehl, um das neue TLS-Geheimnis mit dem privaten Schlüssel und den Zertifikatdateien aus der CA zu erstellen und die Argumente in Klammern <> durch die entsprechenden Informationen zu ersetzen:

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

2. Verwenden Sie den folgenden Befehl und das folgende Beispiel, um die Cluster-Datei CRD (Custom Resource Definition) zu bearbeiten und die zu ändern `spec.selfSigned` Mehrwert für `spec.ca.secretName` So verweisen Sie auf das zuvor erstellte TLS-Geheimnis:

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

CRD:

```
#spec:
#  selfSigned: {}

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

3. Überprüfen Sie mit den folgenden Befehlen und der Beispiel-Ausgabe, ob die Änderungen korrekt sind und das Cluster bereit ist, Zertifikate zu validieren, und ersetzen Sie sie `<ACC-deployment-namespace>` Mit dem Astra Control Center Deployment Namespace:

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

Antwort:

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

4. Erstellen Sie die `certificate.yaml` Datei anhand des folgenden Beispiels, Ersetzen der Platzhalterwerte in Klammern <> durch entsprechende Informationen:

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

5. Erstellen Sie das Zertifikat mit dem folgenden Befehl:

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

6. Überprüfen Sie mithilfe der folgenden Befehl- und Beispielausgabe, ob das Zertifikat korrekt erstellt wurde und mit den während der Erstellung angegebenen Argumenten (z. B. Name, Dauer, Verlängerungsfrist und DNS-Namen).

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

   Antwort:

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

7. Bearbeiten Sie das TLS speichert CRD, um mit dem folgenden Befehl und Beispiel auf Ihren neuen geheimen Zertifikatnamen zu verweisen. Ersetzen Sie die Platzhalterwerte in Klammern <> durch die entsprechenden Informationen

```
kubectl edit tlsstores.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
spec:
  defaultCertificate:
    secretName: <certificate-secret-name>
```

8. Bearbeiten Sie die Option Ingress CRD TLS, um mit dem folgenden Befehl und Beispiel auf Ihr neues Zertifikatgeheimnis zu verweisen und die Platzhalterwerte in Klammern <> durch entsprechende Informationen zu ersetzen:

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

CRD:

```
...
 tls:
    secretName: <certificate-secret-name>
```

9. Navigieren Sie mithilfe eines Webbrowsers zur IP-Adresse der Implementierung von Astra Control Center.
10. Vergewissern Sie sich, dass die Zertifikatdetails mit den Details des installierten Zertifikats übereinstimmen.
11. Exportieren Sie das Zertifikat und importieren Sie das Ergebnis in den Zertifikatmanager in Ihrem Webbrowser.

Copyright-Informationen

Copyright © 2025 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers 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, BEISPIELHAFTE 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 IRGENDEINER 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.