Los geht's

Astra Control Center

NetApp
June 06, 2024

Inhalt

Los geht’s

Anforderungen des Astra Control Centers

Prüfen Sie zunächst die Unterstützung für Ihre Kubernetes Cluster, Applikationen, Lizenzen und Webbrowser.

Allgemeine Anforderungen für den Kubernetes-Cluster

Ein Kubernetes-Cluster muss die folgenden allgemeinen Anforderungen erfüllen, damit Sie ihn über das Astra Control Center erkennen und managen können.

- **Image Registry**: Sie benötigen eine bereits vorhandene private Docker-Image-Registry, mit der Sie Astra Control Center-Bilder erstellen können. Sie müssen über die URL der Bildregistrierung verfügen, in der Sie die Bilder hochladen, und Sie müssen die Bilder für die private Container-Registrierung markiert haben.
- **Trident / ONTAP Storage-Konfiguration**: Astra Control Center erfordert, dass Trident Version 21.01 oder 21.04 bereits installiert und konfiguriert ist, um mit NetApp ONTAP Version 9.5 oder neuer als Storage-Backend zu arbeiten. Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden von Trident bereitgestellten ONTAP-Treiber:
  - ontap-nas
  - ontap-nas-Flexgroup
  - ontap-san
  - ontap-san-Ökonomie

Wenn Sie planen, den Kubernetes-Cluster über das Astra Control Center zu verwalten und die Installation des Astra Control Center über das Cluster zu hosten, gelten die folgenden zusätzlichen Anforderungen:

- Die neueste Version von Kubernetes "snapshot-Controller-Komponente" Installiert ist
- Ein Trident "Objekt der Volumesnapshotklasse" Wurde von einem Administrator definiert
- Im Cluster ist eine standardmäßige Kubernetes-Storage-Klasse vorhanden
- Mindestens eine Storage-Klasse ist für die Verwendung von Trident konfiguriert
- Eine Methode zum Zeigen des FQDN des Astra Control Centers auf die externe IP-Adresse des Astra Control Center-Dienstes

OpenShift Cluster

Astra Control Center erfordert eine Red hat OpenShift-Container-Plattform 4.6.8 oder 4.7-Cluster mit Trident-Storage-Klassen, die durch ONTAP 9.5 oder neuer unterstützt werden und folgende Attribute aufweisen:

- Verfügbare ONTAP Storage-Kapazität von mindestens 300 GB
- 3 Controller-Nodes mit jeweils 4 CPU-Kernen, 16 GB RAM und 120 GB verfügbarer Storage
- 3 Worker-Nodes mit mindestens 12 CPU-Kernen, 32 GB RAM und jeweils 50 GB an verfügbarem Storage
- Kubernetes, Version 1.19 oder 1.20
- Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Dienste im OpenShift-Cluster gesendet werden soll
- Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse

Bei diesen Mindestanforderungen wird davon ausgegangen, dass Astra Control Center die einzige Anwendung ist, die auf dem OpenShift-Cluster ausgeführt wird. Wenn auf dem Cluster zusätzliche Applikationen ausgeführt werden, müssen diese Mindestanforderungen entsprechend angepasst werden.

Stellen Sie sicher, dass Ihr Cluster die Mindestanforderungen erfüllt und die Best Practices für Kubernetes befolgt, damit Astra Control Center in Ihrem Kubernetes Cluster hochverfügbar ist.

OpenShift 4.8 wird nicht unterstützt.

Beim Klonen von Apps muss das Astra Control Center es OpenShift ermöglichen, Volumes anzuhängen und das Eigentum an Dateien zu ändern. Daher muss ONTAP so konfiguriert werden, dass Volume-Vorgänge mit den folgenden Befehlen erfolgreich abgeschlossen werden können:

1. `export-policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
2. `export-policy rule modify -policyname default -ruleindex 1 -anon 65534`

Wenn Sie einen zweiten OpenShift 4.6- oder 4.7-Cluster als gemanagte Computing-Ressource hinzufügen möchten, müssen Sie sicherstellen, dass die Trident Volume Snapshot-Funktion aktiviert ist. Siehe den offiziellen Trident "Anweisungen" Um Volume Snapshots mit Trident zu aktivieren und zu testen.

Anforderungen für das Applikationsmanagement

Astra Control Center verfügt über folgende Anforderungen für das App-Management:

- **Lizenzierung**: Sie benötigen eine Astra Control Center-Lizenz, um Apps mit Astra Control Center zu verwalten.
- **Helm 3**: Wenn Sie Helm zum Bereitstellen von Apps verwenden, benötigt Astra Control Center Helm Version 3. Das Management und Klonen von mit Helm 3 bereitgestellten Apps (oder ein Upgrade von Helm 2 auf Helm 3) werden vollständig unterstützt. Mit Helm 2 implementierte Apps werden nicht unterstützt.
- **Operator Management**: Astra Control Center unterstützt keine Apps, die mit Operators Lifecycle Manager (OLM)-fähigen Operatoren oder Operatoren mit Cluster-Scoped bereitgestellt werden.

Zugang zum Internet

Sie sollten feststellen, ob Sie einen externen Zugang zum Internet haben. Falls nicht, sind einige der Funktionen möglicherweise begrenzt, beispielsweise das Empfangen von Monitoring- und Kennzahlendaten von NetApp Cloud Insights oder das Senden von Support-Paketen an die NetApp Support-Website.

Lizenz

Astra Control Center erfordert eine Astra Control Center-Lizenz für die volle Funktionalität. Anfordern einer Evaluierungslizenz oder Volllizenz von NetApp. Ohne Lizenz können Sie Folgendes nicht ausführen:

- Definieren benutzerdefinierter Applikationen

- Snapshots oder Klone vorhandener Applikationen erstellen
- Konfigurieren von Datensicherungsrichtlinien

Wenn Sie das Astra Control Center ausprobieren möchten, können Sie das auch "Verwenden Sie eine 90-Tage-Evaluierungslizenz".

Servicetyp „Load Balancer“ für lokale Kubernetes-Cluster

Astra Control Center verwendet einen Service des Typs "loadbalancer" (svc/Traefik im Astra Control Center Namespace) und erfordert, dass ihm eine zugängliche externe IP-Adresse zugewiesen wird. Für lokale OpenShift-Cluster ist die Nutzung möglich "MetalLB" So weisen Sie dem Dienst automatisch eine externe IP-Adresse zu. In der Konfiguration des internen DNS-Servers sollten Sie den ausgewählten DNS-Namen für Astra Control Center auf die Load-Balanced IP-Adresse verweisen.

Netzwerkanforderungen

Der Cluster, der Astra Control Center hostet, kommuniziert über die folgenden TCP-Ports. Sie sollten sicherstellen, dass diese Ports über beliebige Firewalls zugelassen sind, und Firewalls so konfigurieren, dass jeder HTTPS-ausgehenden Datenverkehr aus dem Astra-Netzwerk zugelassen wird. Einige Ports erfordern eine Konnektivität zwischen dem Cluster, das Astra Control Center hostet, und jedem verwalteten Cluster (sofern zutreffend).

Produkt	Port	Protokoll	Richtung	Zweck
Astra Control Center	443	HTTPS	Eindringen	UI/API-Zugriff - Stellen Sie sicher, dass dieser Port auf beiden Wegen zwischen dem Cluster geöffnet ist, der Astra Control Center hostet, und jedem verwalteten Cluster
Astra Control Center	9090	HTTPS	• Ingress (zum Cluster-Hosting Astra Control Center) • Ausgehenden (zufälliger Port aus der Node-IP-Adresse jedes Worker-Node jedes verwalteten Clusters)	Kennzahlendaten für den Verbraucher - Stellen Sie sicher, dass jeder gemanagte Cluster auf diesen Port im Cluster-Hosting Astra Control Center zugreifen kann
Trident	34571	HTTPS	Eindringen	Pod-Kommunikation mit Nodes
Trident	9220	HTTP	Eindringen	Endpunkt der Kennzahlen

Unterstützte Webbrowser

Astra Control Center unterstützt aktuelle Versionen von Firefox, Safari und Chrome mit einer Mindestauflösung von 1280 x 720.

Wie es weiter geht

Sehen Sie sich die an "Schnellstart" Überblick.

Schnellstart für Astra Control Center

Diese Seite bietet einen Überblick über die Schritte, die für den Einstieg in das Astra Control Center erforderlich sind. Die Links in den einzelnen Schritten führen zu einer Seite, die weitere Details enthält.

Probieren Sie es aus! Wenn Sie Astra Control Center ausprobieren möchten, können Sie eine 90-Tage-Evaluierungslizenz verwenden. Siehe "Lizenzierungsinformationen" Entsprechende Details.

Kubernetes-Cluster-Anforderungen prüfen

- Astra arbeitet mit Kubernetes-Clustern mit einem in Trident konfigurierten ONTAP-Storage-Back-End.
- Cluster müssen in einem ordnungsgemäßen Zustand mit mindestens drei Online-Worker-Nodes ausgeführt werden.
- Der Cluster muss Kubernetes ausführen.

"Erfahren Sie mehr über die Anforderungen des Astra Control Centers".

Laden Sie Astra Control Center herunter und installieren Sie es

- Laden Sie das Astra Control Center von der NetApp Support-Website herunter.
- Installieren Sie Astra Control Center in Ihrer lokalen Umgebung.
- Trident-Konfiguration wird durch das ONTAP Storage-Back-End unterstützt.

  In unserer ersten Version installieren Sie die Bilder in einer OpenShift-Registrierung oder nutzen Ihre lokale Registrierung.

"Erfahren Sie mehr über die Installation von Astra Control Center".

Führen Sie einige erste Setup-Aufgaben aus

- Fügen Sie eine Lizenz hinzu.
- Ein Kubernetes Cluster hinzufügen und Astra Control Center erkennt Details.
- Fügen Sie ein ONTAP-Storage-Back-End hinzu.
- Optional können Sie einen Objektspeicher-Bucket hinzufügen, der Ihre Applikations-Backups speichert.

"Erfahren Sie mehr über die Ersteinrichtung".

4 Nutzen Sie Das Astra Control Center

Nachdem Sie das Astra Control Center eingerichtet haben, können Sie folgende Schritte ausführen:

- Eine App verwalten. "Erfahren Sie mehr über das Verwalten von Apps".
- Optional können Sie eine Verbindung zu NetApp Cloud Insights herstellen, um Kennzahlen zum Zustand von System, Kapazität und Durchsatz innerhalb der Astra Control Center UI anzuzeigen. "Erfahren Sie mehr über die Verbindung mit Cloud Insights".

5 Fahren Sie mit dieser Schnellstartanleitung fort

"Installieren Sie Astra Control Center".

Weitere Informationen

- "Verwenden Sie die Astra API"

Installieren Sie Astra Control Center

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

- Installieren Sie Astra Control Center
- Melden Sie sich in der UI des Astra Control Center an

Installieren Sie Astra Control Center

Laden Sie zum Installieren des Astra Control Center das Installationspaket von der NetApp Support Site herunter und führen Sie eine Reihe von Befehlen durch, um den Astra Control Center Operator und das Astra Control Center in Ihrer Umgebung zu installieren. Mit diesem Verfahren können Sie Astra Control Center in Internet-angeschlossenen oder luftgekapderten Umgebungen installieren.

Was Sie benötigen

- "Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor".
- Stellen Sie in Ihrem OpenShift-Cluster sicher, dass sich alle Clusterbetreiber in einem ordnungsgemäßen Zustand befinden (`available` Ist `true`):

```
oc get clusteroperators
```

- Stellen Sie in Ihrem OpenShift-Cluster sicher, dass alle API-Services in einem ordnungsgemäßen Zustand sind (`available` Ist `true`):

```
oc get apiservices
```

Über diese Aufgabe

Der Astra Control Center-Installationsprozess führt Folgendes aus:

- Installiert die Astra-Komponenten im `netapp-acc` (Oder benutzerdefinierter Name) Namespace
- Erstellt ein Standardkonto.
- Richtet eine Standard-E-Mail-Adresse für Administratorbenutzer und ein Standardpasswort für ein `ACC-<UUID_of_installation>` Für dieses Beispiel des Astra Control Center. Diesem Benutzer wird die Owner-Rolle im System zugewiesen und ist für die erste Anmeldung bei der UI erforderlich.
- Hilft Ihnen bei der Ermittlung, dass alle Astra Control Center-Pods ausgeführt werden.
- Installiert die Astra UI

Podman-Befehle können anstelle von Docker-Befehlen verwendet werden, wenn Sie das Podman-Repository von Red hat verwenden.

Schritte

1. Laden Sie das Astra Control Center Bundle herunter (`astra-control-center-[version].tar.gz`) Vom "NetApp Support Website".
2. Laden Sie den Zip der Astra Control Center Zertifikate und Schlüssel von herunter "NetApp Support Website".
3. (Optional) Überprüfen Sie mit dem folgenden Befehl die Signatur des Pakets:

```
openssl dgst -sha256 -verify astra-control-center[version].pub
-signature <astra-control-center[version].sig astra-control-
center[version].tar.gz
```

4. Extrahieren Sie die Bilder:

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

5. Wechseln Sie in das Astra-Verzeichnis.

```
cd astra-control-center-[version]
```

6. Fügen Sie die Dateien im Astra Control Center-Bildverzeichnis Ihrer lokalen Registrierung hinzu.

   Siehe ein Beispielskript für das automatische Laden von Bildern unten.

   a. Melden Sie sich bei Ihrer Docker Registrierung an:

   ```
   docker login [Docker_registry_path]
   ```

   b. Laden Sie die Images in Docker.

   c. Markieren Sie die Bilder.

   d. Übertragen Sie die Bilder in Ihre lokale Registrierung.

```
export REGISTRY=[Docker_registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming
the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

7. (Nur bei Registrierung mit auth Anforderungen) Wenn Sie eine Registrierung verwenden, die eine Authentifizierung erfordert, müssen Sie Folgendes tun:

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

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

      Antwort:

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

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

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

      Beispielantwort:

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

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

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

      Beispielantwort:

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

d. Erstellen Sie ein Geheimnis für das `netapp-acc` (Oder benutzerdefinierter Name) 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] --docker-server=[Docker_registry_path] --docker
-username=[username] --docker-password=[token]
```

Antwort

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

8. 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
```

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 `[Docker_registry_path]` Für das `kube-rbac-prox` Bild zum Registrierungspfad, in dem Sie die Bilder in einem vorherigen Schritt verschoben haben.

c. Ändern `[Docker_registry_path]` Für das `acc-operator-controller-manager` Bild zum Registrierungspfad, in dem Sie die Bilder in einem vorherigen Schritt verschoben haben.

```
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
  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: [Docker_registry_path]/kube-rbac-proxy:v0.5.0
        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
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [Docker_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

9. Bearbeiten Sie die Datei Astra Control Center Custom Resource (CR) (`astra_control_center_min.yaml`):

```
vim astra_control_center_min.yaml
```

Falls für Ihre Umgebung zusätzliche Anpassungen erforderlich sind, können Sie dies verwenden `astra_control_center.yaml` Als Alternative CR. `astra_control_center_min.yaml` Ist die Standard-CR und ist für die meisten Installationen geeignet.

Die vom CR konfigurierten Eigenschaften können nach der ersten Implementierung des Astra Control Center nicht geändert werden.

a. Ändern `[Docker_registry_path]` Zum Registrierungspfad, in dem Sie die Bilder im vorherigen Schritt verschoben haben.

b. Ändern Sie das `accountName` Zeichenfolge an den Namen, den Sie dem Konto zuordnen möchten.

c. Ändern Sie das `astraAddress` Zeichenfolge an den FQDN, den Sie in Ihrem Browser für den Zugriff auf Astra verwenden möchten. Verwenden Sie es nicht `http://` Oder `https://` In der Adresse. Kopieren Sie diesen FQDN zur Verwendung in einem Später Schritt.

d. Ändern Sie das `email` Zeichenfolge zur standardmäßigen ursprünglichen Administratoradresse. Kopieren Sie diese E-Mail-Adresse zur Verwendung in A Später Schritt.

e. Ändern `enrolled` Für AutoSupport bis `false` Für Websites ohne Internetverbindung oder Aufbewahrung `true` Für verbundene Standorte.

f. (Optional) Geben Sie einen Vornamen ein `firstName` Und Nachname `lastName` Des Benutzers, der dem Konto zugeordnet ist. Sie können diesen Schritt jetzt oder später in der Benutzeroberfläche ausführen.

g. (Optional) Ändern Sie den `storageClass` Nutzen Sie bei Bedarf für Ihre Installation einen anderen Trident Storage Class-Mitarbeiter.

h. Wenn Sie keine Registrierung verwenden, für die eine Autorisierung erforderlich ist, löschen Sie das `secret` Linie.

```
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: "[Docker_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

10. Installieren Sie den Astra Control Center-Operator:

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

Beispielantwort:

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

11. 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]
```

Beispielantwort:

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

12. Führen Sie den folgenden Patch aus, um ihn zu korrigieren "Bindung der Cluster-Rolle".
13. Installieren Sie das Astra Control Center im `netapp-acc` (Oder Ihr individueller) Namespace:

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

Beispielantwort:

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

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

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

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

Beispielantwort:

```
NAME                                    READY   STATUS    RESTARTS
AGE
acc-helm-repo-5fdfff786f-gkv6z          1/1     Running   0
4m58s
activity-649f869bf7-jn5gs               1/1     Running   0
3m14s
asup-79846b5fdc-s9s97                   1/1     Running   0
3m10s
authentication-84c78f5cf4-qhx9t         1/1     Running   0
118s
billing-9b8496787-v8rzv                 1/1     Running   0
2m54s
bucketservice-5fb876d9d5-wkfvz          1/1     Running   0
3m26s
cloud-extension-f9f4f59c6-dz6s6         1/1     Running   0
3m
```

```
cloud-insights-service-5676b8c6d4-6q7lv      1/1     Running   0
2m52s
composite-compute-7dcc9c6d6c-lxdr6           1/1     Running   0
2m50s
composite-volume-74dbfd7577-cd42b            1/1     Running   0
3m2s
credentials-75dbf46f9d-5qm2b                 1/1     Running   0
3m32s
entitlement-6cf875cb48-gkvhp                 1/1     Running   0
3m12s
features-74fd97bb46-vss2n                    1/1     Running   0
3m6s
fluent-bit-ds-2g9jb                          1/1     Running   0
113s
fluent-bit-ds-5tg5h                          1/1     Running   0
113s
fluent-bit-ds-qfxb8                          1/1     Running   0
113s
graphql-server-7769f98b86-p4qrv              1/1     Running   0
90s
identity-566c566cd5-ntfj6                    1/1     Running   0
3m16s
influxdb2-0                                  1/1     Running   0
4m43s
krakend-5cb8d56978-44q66                     1/1     Running   0
93s
license-66cbbc6f48-27kgf                     1/1     Running   0
3m4s
login-ui-584f7fd84b-dmdrp                    1/1     Running   0
87s
loki-0                                       1/1     Running   0
4m44s
metrics-ingestion-service-6dcfddf45f-mhnvh   1/1     Running   0
3m8s
monitoring-operator-78d67b4d4-nxs6v          2/2     Running   0
116s
nats-0                                       1/1     Running   0
4m40s
nats-1                                       1/1     Running   0
4m26s
nats-2                                       1/1     Running   0
4m15s
nautilus-9b664bc55-rn9t8                     1/1     Running   0
2m56s
openapi-dc5ddfb7d-6q8vh                      1/1     Running   0
3m20s
```

```
polaris-consul-consul-5tjs7                  1/1     Running   0
4m43s
polaris-consul-consul-5wbnx                  1/1     Running   0
4m43s
polaris-consul-consul-bfvl7                  1/1     Running   0
4m43s
polaris-consul-consul-server-0               1/1     Running   0
4m43s
polaris-consul-consul-server-1               1/1     Running   0
4m43s
polaris-consul-consul-server-2               1/1     Running   0
4m43s
polaris-mongodb-0                            2/2     Running   0
4m49s
polaris-mongodb-1                            2/2     Running   0
4m22s
polaris-mongodb-arbiter-0                    1/1     Running   0
4m49s
polaris-ui-6648875998-75d98                  1/1     Running   0
92s
polaris-vault-0                              1/1     Running   0
4m41s
polaris-vault-1                              1/1     Running   0
4m41s
polaris-vault-2                              1/1     Running   0
4m41s
storage-backend-metrics-69546f4fc8-m7lfj     1/1     Running   0
3m22s
storage-provider-5d46f755b-qfv89             1/1     Running   0
3m30s
support-5dc579865c-z4pwq                     1/1     Running   0
3m18s
telegraf-ds-4452f                            1/1     Running   0
113s
telegraf-ds-gnqxl                            1/1     Running   0
113s
telegraf-ds-jhw74                            1/1     Running   0
113s
telegraf-rs-gg6m4                            1/1     Running   0
113s
telemetry-service-6dcc875f98-zft26           1/1     Running   0
3m24s
tenancy-7f7f77f699-q7l6w                     1/1     Running   0
3m28s
traefik-769d846f9b-c9crt                     1/1     Running   0
83s
```

```
traefik-769d846f9b-l9n4k                          1/1     Running   0
67s
trident-svc-8649c8bfc5-pdj79                      1/1     Running   0
2m57s
vault-controller-745879f98b-49c5v                 1/1     Running   0
4m51s
```

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

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

16. Wenn alle Pods ausgeführt werden, überprüfen Sie den Installationserfolg, indem Sie die AstraControlCenter-Instanz abrufen, die vom ACC-Operator installiert wurde.

```
kubectl get acc -o yaml -n netapp-acc
```

17. Prüfen Sie die `status.deploymentState` Feld in der Antwort für das `Deployed` Wert: Wenn die Bereitstellung nicht erfolgreich war, wird stattdessen eine Fehlermeldung angezeigt.

> Sie werden die verwenden `uuid` Im nächsten Schritt.

```
apiVersion: v1
items:
- apiVersion: astra.netapp.io/v1
  kind: AstraControlCenter
  metadata:
    creationTimestamp: "2021-07-28T21:36:49Z"
    finalizers:
    - astracontrolcenter.netapp.io/finalizer
   generation: 1
    name: astra
    namespace: netapp-acc
    resourceVersion: "27797604"
    selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
    uid: 61cd8b65-047b-431a-ba35-510afcb845f1
  spec:
    accountName: Example
    astraAddress: astra.example.com
    astraResourcesScaler: "Off"
    astraVersion: 21.08.52
    autoSupport:
      enrolled: false
    email: admin@example.com
    firstName: SRE
    lastName: Admin
    imageRegistry:
      name: registry_name/astra
  status:
    certManager: deploy
    deploymentState: Deployed
    observedGeneration: 1
    observedVersion: 21.08.52
    postInstall: Complete
    uuid: c49008a5-4ef1-4c5d-a53e-830daf994116
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

18. Um das einmalige Passwort zu erhalten, das Sie bei der Anmeldung beim Astra Control Center verwenden, kopieren Sie das `status.uuid` Wert aus der Antwort im vorherigen Schritt. Das Passwort lautet `ACC-` Anschließend der UUID-Wert (`ACC-[UUID]` Oder in diesem Beispiel `ACC-c49008a5-4ef1-4c5d-a53e-830daf994116`).

Melden Sie sich in der UI des Astra Control Center an

Nach der Installation von ACC ändern Sie das Kennwort für den Standardadministrator und melden sich beim Dashboard von ACC UI an.

Schritte

1. Geben Sie in einem Browser den FQDN ein, den Sie in verwendet haben `astraAddress` Im `astra_control_center_min.yaml` CR, wenn Sie haben ACC 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_min.yaml` CR, wenn Sie haben ACC installiert, Gefolgt von dem Einzeitkennwort (`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 es sich um Ihre erste Anmeldung handelt und Sie das Passwort vergessen haben und noch keine anderen Administratorkonten erstellt wurden, wenden Sie sich an den NetApp Support, um Unterstützung bei der Passwortwiederherstellung zu erhalten.

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

Beheben Sie die Fehlerbehebung für die Installation

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

Schritte

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

```
kubectl logs --follow -n netapp-acc-operator $(kubectl get pods -n
netapp-acc-operator -o name)  -c manager
```

Wie es weiter geht

Führen Sie die Implementierung durch "Setup-Aufgaben".

Einrichten des Astra Control Center

Nach der Installation von Astra Control Center, melden Sie sich in der UI an und ändern Sie Ihr Passwort, Sie möchten eine Lizenz einrichten, Cluster hinzufügen, Speicher verwalten und Buckets hinzufügen.

Aufgaben

- Fügen Sie eine Lizenz für Astra Control Center hinzu
- Cluster hinzufügen
- Fügen Sie ein Storage-Back-End hinzu
- Fügen Sie einen Bucket hinzu

Fügen Sie eine Lizenz für Astra Control Center hinzu

Sie können eine neue Lizenz über die UI oder hinzufügen "API" Um die Funktionalität des Astra Control Center voll zu nutzen. Ohne Lizenz ist Ihre Verwendung von Astra Control Center auf das Management von Benutzern und das Hinzufügen neuer Cluster beschränkt.

Was Sie benötigen

Wenn Sie Astra Control Center von heruntergeladen "NetApp Support Website", Sie haben auch die NetApp Lizenzdatei (NLF) heruntergeladen. Stellen Sie sicher, dass Sie Zugriff auf diese Lizenzdatei haben.

Informationen zum Aktualisieren einer vorhandenen Testversion oder Volllizenz finden Sie unter "Aktualisieren einer vorhandenen Lizenz".

Fügen Sie eine vollständige Lizenz oder eine Evaluierungslizenz hinzu

Astra Control Center Lizenzen messen die CPU-Ressourcen mithilfe von Kubernetes CPU-Einheiten. Die Lizenz muss die CPU-Ressourcen berücksichtigen, die den Worker-Nodes aller verwalteten Kubernetes-Cluster zugewiesen sind. Bevor Sie eine Lizenz hinzufügen, müssen Sie die Lizenzdatei (NLF) vom beziehen "NetApp Support Website".

Sie können das Astra Control Center auch mit einer Evaluierungslizenz ausprobieren, mit der Sie das Astra Control Center 90 Tage ab dem Tag, an dem Sie die Lizenz herunterladen, nutzen können. Sie können sich durch die Anmeldung für eine kostenlose Testversion anmelden "Hier".

Wenn Ihre Installation die Anzahl der lizenzierten CPU-Einheiten überschreitet, verhindert Astra Control Center die Verwaltung neuer Anwendungen. Bei Überschreitung der Kapazität wird eine Meldung angezeigt.

Schritte

1. Melden Sie sich in der UI des Astra Control Center an.
2. Wählen Sie **Konto** > **Lizenz**.
3. Wählen Sie **Lizenz Hinzufügen**.
4. Rufen Sie die Lizenzdatei (NLF) auf, die Sie heruntergeladen haben.
5. Wählen Sie **Lizenz Hinzufügen**.

Auf der Seite **Konto** > **Lizenz** werden Lizenzinformationen, Ablaufdatum, Lizenzseriennummer, Konto-ID und verwendete CPU-Einheiten angezeigt.

Wenn Sie über eine Evaluierungslizenz verfügen, sollten Sie Ihre Konto-ID speichern, um Datenverlust im Falle eines Ausfalls des Astra Control Center zu vermeiden, wenn Sie ASUPs nicht senden.

Cluster hinzufügen

Zum Management von Applikationen fügen Sie einen Kubernetes-Cluster hinzu und managen ihn als Computing-Ressource. Um Ihre Kubernetes-Applikationen zu ermitteln, müssen Sie einen Cluster hinzufügen, in dem Astra Control Center ausgeführt werden kann.

Wir empfehlen, dass Astra Control Center den Cluster, der zuerst bereitgestellt wird, verwaltet, bevor Sie zum Management weitere Cluster zum Astra Control Center hinzufügen. Das Management des anfänglichen Clusters ist erforderlich, um Kubemetrics-Daten und Cluster-zugeordnete Daten zur Metriken und Fehlerbehebung zu senden. Sie können die * Cluster hinzufügen* Funktion verwenden, um einen Cluster mit Astra Control Center zu verwalten.

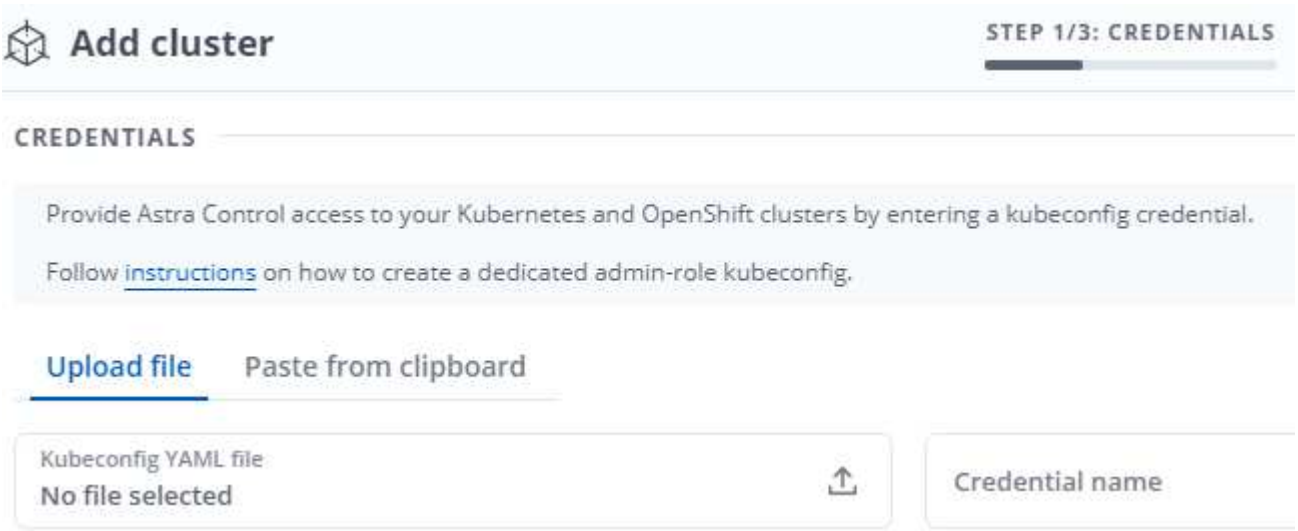
Was Sie und#8217;ll benötigen

Bevor Sie ein Cluster hinzufügen, überprüfen und führen Sie die erforderlichen Maßnahmen durch "Erforderliche Aufgaben".

Schritte

1. Wählen Sie in der Astra Control Center-Benutzeroberfläche auf dem Dashboard * im Bereich Cluster die Option **Add** aus.
2. Laden Sie im Fenster **Cluster hinzufügen** ein `kubeconfig.yaml` Datei oder fügen Sie den Inhalt eines ein `kubeconfig.yaml` Datei:

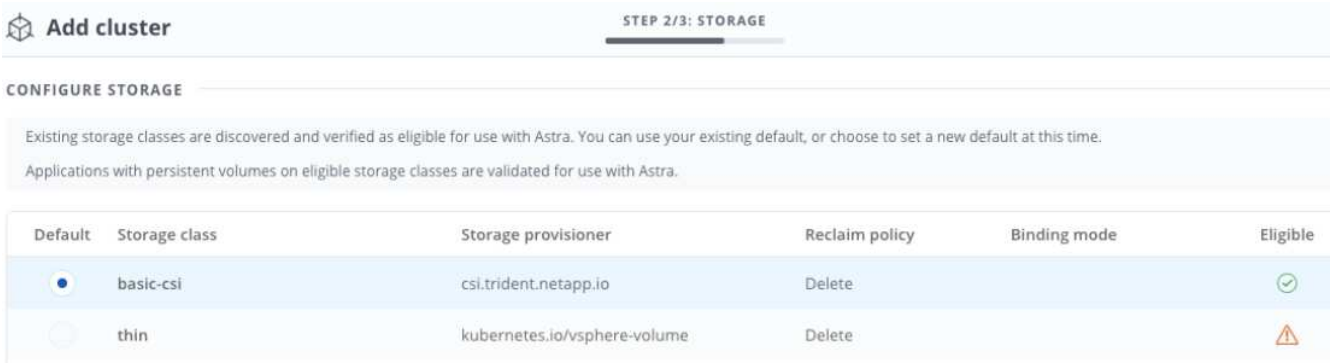
   Der `kubeconfig.yaml` Die Datei sollte **nur die Cluster-Anmeldedaten für einen Cluster** enthalten.



   Wenn Sie Ihre eigenen erstellen `kubeconfig` Datei, Sie sollten nur ein **ein**-Kontext -Element darin definieren. Siehe "Kubernetes-Dokumentation" Weitere Informationen zum Erstellen `kubeconfig` Dateien:

3. Geben Sie einen Namen für die Anmeldeinformationen an. Standardmäßig wird der Name der Anmeldeinformationen automatisch als Name des Clusters ausgefüllt.
4. Wählen Sie * Storage konfigurieren* aus.
5. Wählen Sie die Storage-Klasse aus, die für diesen Kubernetes-Cluster verwendet werden soll, und wählen Sie **Review** aus.

Sie sollten sich für eine von ONTAP Storage gesicherte Trident Storage-Klasse entscheiden.



6. Überprüfen Sie die Informationen, und wenn alles gut aussieht, wählen Sie **Cluster hinzufügen**.

Ergebnis

Der Cluster wechselt in den **Entdeckungs**-Status und dann in **running**. Sie haben erfolgreich ein Kubernetes-Cluster hinzugefügt und verwalten es jetzt im Astra Control Center.

Nachdem Sie einen Cluster hinzugefügt haben, der im Astra Control Center verwaltet werden soll, kann es in einigen Minuten dauern, bis der Monitoring-Operator implementiert ist. Bis dahin wird das Benachrichtigungssymbol rot und ein Ereignis **Überwachung Agent-Status-Prüfung fehlgeschlagen** protokolliert. Sie können dies ignorieren, da das Problem gelöst wird, wenn Astra Control Center den richtigen Status erhält. Wenn sich das Problem in wenigen Minuten nicht beheben lässt, wechseln Sie zum Cluster und führen Sie aus `oc get pods -n netapp-monitoring` Als Ausgangspunkt. Um das Problem zu beheben, müssen Sie sich die Protokolle des Überwachungsperbers ansehen.

Fügen Sie ein Storage-Back-End hinzu

Sie können ein Storage-Backend hinzufügen, sodass Astra Control die Ressourcen managen kann. Durch das Management von Storage-Clustern in Astra Control als Storage-Backend können Sie Verbindungen zwischen persistenten Volumes (PVS) und dem Storage-Backend sowie zusätzliche Storage-Kennzahlen abrufen.

Es gibt folgende Möglichkeiten, ein Storage-Backend hinzuzufügen:

- Konfigurieren Sie Storage, wenn Sie ein Cluster hinzufügen. Siehe "Cluster hinzufügen".
- Fügen Sie mit der Option Dashboard oder Back-Ends ein ermittelte Speicher-Backend hinzu.

Sie können ein bereits ermittelte Speicher-Backend mit folgenden Optionen hinzufügen:

- Fügen Sie über Dashboard ein Storage-Backend hinzu
- Fügen Sie mit der Option Back-Ends Speicher-Backend hinzu

Fügen Sie über Dashboard ein Storage-Backend hinzu

1. Führen Sie im Dashboard einen der folgenden Schritte aus:
   a. Wählen Sie im Bereich Dashboard Storage Backend die Option **Verwalten** aus.
   b. Wählen Sie im Abschnitt Dashboard-Ressourcen-Übersicht > Storage-Back-Ends die Option **Hinzufügen** aus.

2. Geben Sie die Anmeldedaten für den ONTAP-Administrator ein, und wählen Sie **Überprüfen**.
3. Bestätigen Sie die Backend-Details und wählen Sie **Verwalten**.

   Das Backend wird in der Liste mit Zusammenfassungsinformationen angezeigt.

Fügen Sie mit der Option Back-Ends Speicher-Backend hinzu

1. Wählen Sie im linken Navigationsbereich **Backend** aus.
2. Wählen Sie **Verwalten**.
3. Geben Sie die Anmeldedaten für den ONTAP-Administrator ein, und wählen Sie **Überprüfen**.
4. Bestätigen Sie die Backend-Details und wählen Sie **Verwalten**.

   Das Backend wird in der Liste mit Zusammenfassungsinformationen angezeigt.

5. Um Details zum Back-End-Speicher anzuzeigen, wählen Sie ihn aus.

Es werden auch persistente Volumes angezeigt, die von Applikationen im gemanagten Computing-Cluster verwendet werden.

Fügen Sie einen Bucket hinzu

Das Hinzufügen von Objektspeicher-Bucket-Providern ist wichtig, wenn Sie Ihre Applikationen und Ihren persistenten Storage sichern möchten oder Applikationen über Cluster hinweg klonen möchten. Astra Control speichert diese Backups oder Klone in den von Ihnen definierten Objektspeicher-Buckets.

Wenn Sie einen Bucket hinzufügen, markiert Astra Control einen Bucket als Standard-Bucket-Indikator. Der erste von Ihnen erstellte Bucket wird der Standard-Bucket.

Sie brauchen keinen Eimer, wenn Sie Ihre Anwendungskonfiguration und Ihren persistenten Storage im selben Cluster klonen.

Verwenden Sie einen der folgenden Bucket-Typen:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Allgemein S3

Obwohl Astra Control Center Amazon S3 als Generic S3 Bucket-Provider unterstützt, unterstützt Astra Control Center möglicherweise nicht alle Objektspeicher-Anbieter, die die S3-Unterstützung von Amazon beanspruchen.

Anweisungen zum Hinzufügen von Buckets mithilfe der Astra API finden Sie unter "Astra Automation und API-Informationen".

Schritte

1. Wählen Sie im linken Navigationsbereich **Buckets** aus.
   a. Wählen Sie **Hinzufügen**.
   b. Wählen Sie den Bucket-Typ aus.

Wenn Sie einen Bucket hinzufügen, wählen Sie den richtigen Bucket-Provider-Typ mit den Zugangsdaten aus, die für diesen Provider korrekt sind. Die UI akzeptiert beispielsweise NetApp ONTAP S3 als Typ mit StorageGRID Zugangsdaten. Dies führt jedoch dazu, dass alle künftigen Applikations-Backups und -Wiederherstellungen mit diesem Bucket fehlschlagen.

c. Erstellen Sie einen neuen Bucket-Namen oder geben Sie einen vorhandenen Bucket-Namen und eine optionale Beschreibung ein.

Der Bucket-Name und die Beschreibung erscheinen als Backup-Speicherort, den Sie später wählen können, wenn Sie ein Backup erstellen. Der Name wird auch während der Konfiguration der Schutzrichtlinien angezeigt.

d. Geben Sie den Namen oder die IP-Adresse des S3-Servers ein.

e. Wenn dieser Bucket der Standard-Bucket für alle Backups sein soll, prüfen Sie den `Make this bucket the default bucket for this private cloud` Option.

Diese Option wird nicht für den ersten von Ihnen erstellten Bucket angezeigt.

f. Mit Hinzufügen fortfahren Anmeldeinformationen.

Fügen Sie S3-Zugriffsdaten hinzu

Fügen Sie Ihre Zugangsdaten für S3-Zugriff jederzeit hinzu.

Schritte

1. Wählen Sie im Dialogfeld Buckets entweder die Registerkarte **Hinzufügen** oder **vorhandene verwenden** aus.
   a. Geben Sie einen Namen für die Anmeldedaten ein, der sie von anderen Anmeldeinformationen in Astra Control unterscheidet.
   b. Geben Sie die Zugriffs-ID und den geheimen Schlüssel ein, indem Sie den Inhalt aus der Zwischenablage einfügen.

Was kommt als Nächstes?

Nachdem Sie sich angemeldet haben und Cluster zum Astra Control Center hinzugefügt haben, können Sie die Anwendungsdatenmanagement-Funktionen von Astra Control Center nutzen.

- "Benutzer managen"
- "Starten Sie das Anwendungsmanagement"
- "Schützen von Applikationen"
- "Applikationen klonen"
- "Benachrichtigungen verwalten"
- "Verbinden Sie sich mit Cloud Insights"
- "Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu"

Weitere Informationen

- "Verwenden Sie die Astra API"
- "Bekannte Probleme"

Voraussetzungen für das Hinzufügen eines Clusters

Sie sollten sicherstellen, dass die Voraussetzungen erfüllt sind, bevor Sie ein Cluster hinzufügen. Außerdem sollten Sie die Eignungskontrollen durchführen, um sicherzustellen, dass Ihr Cluster zum Astra Control Center hinzugefügt werden kann.

Was benötigen Sie vor dem Hinzufügen eines Clusters

- Cluster mit OpenShift 4.6 oder 4.7, wobei Trident StorageClasses auf ONTAP 9.5 oder höher unterstützt werden
  - Ein oder mehrere Worker-Nodes mit mindestens 1 GB RAM für laufende Telemetrieservices verfügbar.

> Wenn Sie planen, als gemanagte Computing-Ressource einen zweiten OpenShift 4.6- oder 4.7-Cluster hinzuzufügen, sollten Sie sicherstellen, dass die Trident Volume Snapshot-Funktion aktiviert ist. Siehe den offiziellen Trident "Anweisungen" Um Volume Snapshots mit Trident zu aktivieren und zu testen.

- Der Superuser und die Benutzer-ID, die auf dem ONTAP-System für die Sicherung und Wiederherstellung von Apps mit Astra Control Center (ACC) eingestellt sind. Führen Sie die folgenden Befehle in der ONTAP-Befehlszeile aus:
  `export policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
  `export-policy rule modify -policyname default -ruleindex 1 -anon 65534` (Dies ist der Standardwert)

Führen Sie Eignungsprüfungen durch

Führen Sie die folgenden Eignungsprüfungen durch, um sicherzustellen, dass Ihr Cluster zum Astra Control Center hinzugefügt werden kann.

Schritte

1. Überprüfen Sie die Trident Version.

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

Wenn Trident vorhanden ist, wird eine Ausgabe ähnlich der folgenden ausgegeben:

```
NAME      VERSION
trident   21.04.0
```

Wenn Trident nicht vorhanden ist, wird eine Ausgabe wie die folgende angezeigt:

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

Wenn Trident nicht installiert ist oder die installierte Version nicht die neueste ist, müssen Sie die neueste Version von Trident installieren, bevor Sie fortfahren. Siehe "Trident Dokumentation" Weitere Anweisungen.

2. Prüfen Sie, ob die Storage-Klassen die unterstützten Trident Treiber verwenden. Der bereitstellungsname sollte lauten `csi.trident.netapp.io`. Das folgende Beispiel zeigt:

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

Erstellen Sie ein „admin-Role“-kubeconfig

Stellen Sie sicher, dass Sie die folgenden Schritte auf Ihrem Gerät ausführen:

- `kubectl` v1.19 oder höher installiert
- Ein aktiver kubeconfig mit Clusteradministratorrechten für den aktiven Kontext

Schritte

1. Erstellen Sie ein Service-Konto wie folgt:

   a. Erstellen Sie eine Dienstkontendatei mit dem Namen `astracontrol-service-account.yaml`.

   Passen Sie Namen und Namespace nach Bedarf an. Wenn hier Änderungen vorgenommen werden, sollten Sie die gleichen Änderungen in den folgenden Schritten anwenden.

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

+

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

   a. Wenden Sie das Servicekonto an:

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

2. Gewähren Sie Cluster-Admin-Berechtigungen wie folgt:

   a. Erstellen Sie ein `ClusterRoleBinding` Datei aufgerufen `astracontrol-clusterrolebinding.yaml`.

      Passen Sie bei Bedarf alle beim Erstellen des Dienstkontos geänderten Namen und Namespaces an.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

+

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

   a. Wenden Sie die Bindung der Cluster-Rolle an:

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

3. Listen Sie die Geheimnisse des Dienstkontos auf, ersetzen Sie `<context>` Mit dem richtigen Kontext für Ihre Installation:

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

   Das Ende der Ausgabe sollte wie folgt aussehen:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

   Die Indizes für jedes Element im `secrets` Array beginnt mit 0. Im obigen Beispiel der Index für `astracontrol-service-account-dockercfg-vhz87` Wäre 0 und der Index für `astracontrol-`

`service-account-token-r59kr` Sind es 1. Notieren Sie in Ihrer Ausgabe den Index für den Namen des Dienstkontos, der das Wort „Token“ darin enthält.

4. Erzeugen Sie den kubeconfig wie folgt:
   a. Erstellen Sie ein `create-kubeconfig.sh` Datei: Wenn der im vorherigen Schritt erwähnte Token-Index nicht 0 war, ersetzen Sie den Wert für `TOKEN_INDEX` Am Anfang des folgenden Skripts mit dem korrekten Wert.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment. Replace the value for
TOKEN_INDEX from
# the output in the previous step if it was not 0. If you didn't
change anything
# else above, don't change anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'
TOKEN_INDEX=0

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
```

```
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. Geben Sie die Befehle an, um sie auf Ihren Kubernetes-Cluster anzuwenden.

```
source create-kubeconfig.sh
```

5. (**Optional**) Umbenennen Sie die kubeconfig in einen aussagekräftigen Namen für Ihren Cluster. Schützen Sie die Cluster-Anmeldedaten.

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

Was kommt als Nächstes?

Jetzt, wo du überprüft hast, dass die Voraussetzungen erfüllt sind, bist du bereit "Fügen Sie einen Cluster hinzu".

Weitere Informationen

- "Trident Dokumentation"
- "Verwenden Sie die Astra API"

Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

Sie können das vorhandene selbst signierte TLS-Zertifikat entfernen und durch ein TLS-Zertifikat ersetzen, das von einer Zertifizierungsstelle (CA) signiert ist.

Was Sie benötigen

- 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

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 ein neues Zertifikat hinzu

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>
....

#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>
....

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

5. 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>
....

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 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.containo.us -n <ACC-deployment-
namespace>
....

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

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

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

Häufig gestellte Fragen zum Astra Control Center

Diese FAQ kann Ihnen helfen, wenn Sie nur nach einer schnellen Antwort auf eine Frage suchen.

Überblick

In den folgenden Abschnitten finden Sie Antworten auf einige zusätzliche Fragen, die Sie bei der Verwendung von Astra Control Center interessieren könnten. Weitere Erläuterungen erhalten Sie von astra.feedback@netapp.com

Zugang zum Astra Control Center

Was ist die Astra Control URL?

Astra Control Center verwendet lokale Authentifizierung und eine spezifische URL für jede Umgebung.

Geben Sie für die URL in einem Browser den vollständig qualifizierten Domänennamen (FQDN) ein, den Sie im Feld spec.astraAddress in der Datei astra_control_Center_min.yaml Custom Resource Definition (CRD) festgelegt haben, wenn Sie Astra Control Center installiert haben. Die E-Mail ist der Wert, den Sie im Feld Spec.email im astra_Control_Center_min.yaml CRD festgelegt haben.

Ich verwende die Evaluierungslizenz. Wie ändere ich die Volllizenz?

Sie können die vollständige Lizenz ganz einfach von der NetApp Lizenzdatei (NetApp License File, NLF)

erhalten.

Schritte

- Wählen Sie in der linken Navigationsleiste **Konto** > **Lizenz**.
- Wählen Sie **Lizenz hinzufügen**.
- Navigieren Sie zu der Lizenzdatei, die Sie heruntergeladen haben, und wählen Sie **Hinzufügen**.

Ich verwende die Evaluierungslizenz. Kann ich trotzdem Apps verwalten?

Ja, Sie können die Funktionalität Apps verwalten mit der Evaluierungslizenz testen.

Kubernetes Cluster werden registriert

Nach dem Hinzufügen von Astra Control müssen ich die Worker-Nodes zu meinem Kubernetes Cluster hinzufügen. Was soll ich tun?

Vorhandenen Pools können neue Worker Nodes hinzugefügt werden. Diese werden automatisch von Astra Control entdeckt. Wenn die neuen Knoten in Astra Control nicht sichtbar sind, prüfen Sie, ob auf den neuen Worker Nodes der unterstützte Bildtyp ausgeführt wird. Sie können den Zustand der neuen Worker-Nodes auch mit überprüfen `kubectl get nodes` Befehl.

Wie entnehme ich einen Cluster richtig?

1. "Lösen Sie die Anwendungen von Astra Control".
2. "Lösen Sie das Cluster über Astra Control".

Was passiert mit meinen Anwendungen und Daten, nachdem ich den Kubernetes Cluster aus Astra Control entfernt habe?

Das Entfernen eines Clusters aus Astra Control führt keine Änderungen an der Cluster-Konfiguration (Applikationen und persistenter Storage) durch. Astra Control Snapshots oder Backups, die von Applikationen auf diesem Cluster erstellt werden, sind zur Wiederherstellung nicht verfügbar. Die von Astra Control erstellten persistenten Storage Backups bleiben innerhalb des Astra Control, sind aber nicht für die Wiederherstellung verfügbar.

> Entfernen Sie immer einen Cluster aus Astra Control, bevor Sie ihn mit anderen Methoden löschen. Das Löschen eines Clusters mithilfe eines anderen Tools, während es noch von Astra Control gemanagt wird, kann zu Problemen mit Ihrem Astra Control Konto führen.

Wird NetApp Trident deinstalliert, wenn ich einen Kubernetes Cluster aus Astra Control entferne?

Trident wird nicht aus einem Cluster deinstalliert, wenn Sie es aus Astra Control entfernen.

Management von Applikationen

Kann Astra Control eine Anwendung bereitstellen?

Astra Control implementiert keine Applikationen. Applikationen müssen außerhalb von Astra Control bereitgestellt werden.

Was passiert mit Anwendungen, nachdem ich sie von Astra Control aus verwaltet habe?

Alle bestehenden Backups oder Snapshots werden gelöscht. Applikationen und Daten sind weiterhin verfügbar. Datenmanagement-Vorgänge stehen nicht für nicht verwaltete Anwendungen oder für Backups oder Snapshots zur Verfügung, die dazu gehören.

Kann Astra Control eine Applikation managen, die sich auf Storage anderer Anbieter befindet?

Nein Astra Control kann zwar Applikationen erkennen, die Storage anderer Anbieter nutzen, aber keine Applikation managen, die Storage von anderen Anbietern verwendet.

Sollte ich Astra Control selbst verwalten? Nein, Sie sollten Astra Control nicht selbst verwalten, weil es sich um eine "System-App" handelt.

Datenmanagement-Vorgänge

Es gibt Schnappschüsse in meinem Konto, die ich nicht erstellt habe. Woher kamen sie?

In manchen Situationen erstellt Astra Control automatisch einen Snapshot im Rahmen eines Backup-, Klon- oder Wiederherstellungsprozesses.

Meine Anwendung verwendet mehrere PVS. Wird Astra Control Snapshots und Backups all dieser VES machen?

Ja. Ein Snapshot-Vorgang auf einer Anwendung von Astra Control umfasst die Momentaufnahme aller VES, die an die VES der Anwendung gebunden sind.

Kann ich die von Astra Control erstellten Snapshots direkt über eine andere Schnittstelle oder Objekt-Storage managen?

Nein Snapshots und Backups von Astra Control können nur mit Astra Control verwaltet werden.