Übersicht über die Installation

Astra Control Center

NetApp
November 21, 2023

This PDF was generated from https://docs.netapp.com/de-de/astra-control-center-2204/get-started/install_acc.html on November 21, 2023. Always check docs.netapp.com for the latest.

Inhalt

Übersicht über die Installation 1
Installieren Sie das Astra Control Center mithilfe des Standardprozesses 1
Installieren Sie Astra Control Center mit OpenShift OperatorHub 23
Installieren Sie Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend 29

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"

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, um Astra Control Center Operator und Astra Control Center in Ihrer Umgebung zu installieren. Mit diesem Verfahren können Sie Astra Control Center in Internet-angeschlossenen oder luftgekapderten Umgebungen installieren.

Für Red hat OpenShift-Umgebungen können Sie auch ein verwenden "Alternativverfahren" So installieren Sie Astra Control Center mithilfe des OpenShift OperatorHub.

Was Sie benötigen

- "Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor".
- Stellen Sie sicher, dass alle Cluster Operator in einem ordnungsgemäßen Zustand und verfügbar sind.

  OpenShift-Beispiel:

  ```
  oc get clusteroperators
  ```

- Stellen Sie sicher, dass alle API-Services in einem ordnungsgemäßen Zustand und verfügbar sind:

  OpenShift-Beispiel:

  ```
  oc get apiservices
  ```

- Der Astra FQDN, den Sie verwenden möchten, muss für diesen Cluster routingfähig sein. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.

Über diese Aufgabe

Der Astra Control Center-Installationsprozess führt Folgendes aus:

- Installiert die Astra-Komponenten im `netapp-acc` (Oder Name des benutzerdefinierten 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

> (Gilt nur für die Version des Astra Data Store Early Access Program (EAP). Wenn Sie den Astra Data Store über das Astra Control Center verwalten und VMware-Workflows aktivieren möchten, implementieren Sie Astra Control Center nur auf dem `pcloud` Und nicht auf dem `netapp-acc` Namespace oder ein benutzerdefinierter Namespace, der in den Schritten dieses Verfahrens beschrieben wird.

> Führen Sie den folgenden Befehl während der gesamten Installation nicht aus, um zu vermeiden, dass alle Astra Control Center Pods gelöscht werden: `kubectl delete -f astra_control_center_operator_deploy.yaml`

> Wenn Sie Podman von Red hat anstelle von Docker Engine verwenden, können Podman-Befehle anstelle von Docker-Befehlen verwendet werden.

Schritte

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

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

Laden Sie das Astra Control Center Bundle herunter und entpacken Sie es aus

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 aus dem 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
```

Installieren Sie das NetApp Astra kubectl Plug-in

Der NetApp Astra `kubectl` Kommandozeilen-Plugin spart Zeit bei der Ausführung von Routineaufgaben im Zusammenhang mit der Bereitstellung und dem Upgrade von Astra Control Center.

Was Sie benötigen

NetApp bietet Binärdateien für das Plug-in für verschiedene CPU-Architekturen und Betriebssysteme. Sie müssen wissen, welche CPU und welches Betriebssystem Sie haben, bevor Sie diese Aufgabe ausführen. Unter Linux- und Mac-Betriebssystemen können Sie die verwenden `uname -a` Befehl zum Sammeln dieser Informationen.

Schritte

1. Nennen Sie den verfügbaren NetApp Astra `kubectl` Plugin-Binärdateien, und notieren Sie den Namen der Datei, die Sie für Ihr Betriebssystem und CPU-Architektur benötigen:

   ```
   ls kubectl-astra/
   ```

2. Kopieren Sie die Datei an denselben Speicherort wie der Standard `kubectl` Utility: In diesem Beispiel ist der `kubectl` Das Dienstprogramm befindet sich im `/usr/local/bin` Verzeichnis. Austausch `<binary-name>` Mit dem Namen der benötigten Datei:

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

Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

1. Wechseln Sie in das Astra-Verzeichnis:

   ```
   cd acc
   ```

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

   > Siehe Beispielskripts zum automatischen Laden von Bildern unten.

   a. Melden Sie sich bei Ihrer Registrierung an:

      Docker:

      ```
      docker login [your_registry_path]
      ```

      Podman:

```
podman login [your_registry_path]
```

b. Verwenden Sie das entsprechende Skript, um die Bilder zu laden, die Bilder zu kennzeichnen, und Schieben Sie die Bilder in Ihre lokale Registrierung:

Docker:

```
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
```

Podman:

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

Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen

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

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

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

Antwort:

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

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

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

Beispielantwort:

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

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

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

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 namespace] --docker-server=[your_registry_path]
--docker-username=[username] --docker-password=[token]
```

Antwort

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

a. (Optional) Wenn Sie möchten, dass der Cluster nach der Installation automatisch vom Astra Control Center verwaltet wird, stellen Sie sicher, dass Sie den kubeconfig als Geheimnis innerhalb des Astra Control Center Namespace angeben, in dem Sie diesen Befehl einsetzen möchten:

```
kubectl create secret generic [acc-kubeconfig-cred or custom secret
name] --from-file=<path-to-your-kubeconfig> -n [netapp-acc or custom
namespace]
```

Installieren Sie den Operator Astra Control Center

1. 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: <name_of_secret_with_creds_to_local_registry>
      ```

   b. Ändern `[your_registry_path]` Für das `kube-rbac-proxy` Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben Vorheriger Schritt.

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

   d. (Für Installationen mit Astra Data Store Vorschau) Siehe dieses bekannte Problem bzgl. "Provisorer der Speicherklasse und zusätzliche Änderungen, die Sie an der YAML vornehmen müssen".

```
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: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

2. 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
```

Konfigurieren Sie Astra Control Center

1. Bearbeiten Sie die Datei Astra Control Center Custom Resource (CR) (`astra_control_center_min.yaml`) Um Konto, AutoSupport, Registrierung und andere notwendige Konfigurationen zu machen:

> 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.

```
vim astra_control_center_min.yaml
```

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

> Wenn Sie eine Registrierung verwenden, für die keine Autorisierung erforderlich ist, müssen Sie das löschen `secret` Zeile in `imageRegistry` Oder die Installation schlägt fehl.

a. Ändern `[your_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. (Optional) Wenn der Cluster nach der Installation automatisch von Astra Control Center verwaltet werden soll und schon vorhanden ist Schuf das Geheimnis, das den kubeconfig für diesen Cluster enthältGeben Sie den Namen des Geheimnisses an, indem Sie dieser YAML-Datei ein neues Feld hinzufügen `astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"`

i. Führen Sie einen der folgenden Schritte aus:

   - **Anderer Ingress-Controller (ingressType:Generic)**: Dies ist die Standard-Aktion mit Astra Control Center. Nachdem Astra Control Center bereitgestellt wurde, müssen Sie den Ingress-Controller so konfigurieren, dass Astra Control Center mit einer URL verfügbar ist.

     Die standardmäßige Astra Control Center-Installation stellt das Gateway ein (`service/traefik`) Vom Typ zu sein `ClusterIP`. Bei dieser Standardinstallation müssen Sie zusätzlich einen Kubernetes ProgressController/Ingress einrichten, um den Datenverkehr dorthin zu leiten. Wenn Sie ein Ingress verwenden möchten, lesen Sie "Eindringen für den Lastenausgleich einrichten".

   - **Service Load Balancer (ingressType:AccTraefik)**: Wenn Sie keinen IngressController installieren oder eine Ingress-Ressource erstellen möchten, stellen Sie ein `ingressType` Bis `AccTraefik`.

     Dies implementiert das Astra Control Center `traefik` Gateway als Service des Typs Kubernetes Load Balancer:

     Astra Control Center nutzt einen Service vom Typ „loadbalancer“ (`svc/traefik` Im Astra Control Center Namespace) und erfordert, dass ihm eine zugängliche externe IP-Adresse zugewiesen wird. Wenn in Ihrer Umgebung Load Balancer zugelassen sind und Sie noch nicht eine 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.

     Einzelheiten zum Servicetyp von „loadbalancer“ und Ingress finden Sie unter "Anforderungen".

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"
  ingressType: "Generic"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

Komplette Astra Control Center und Bedienerinstallation

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

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

   Beispielantwort:

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

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

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

   Beispielantwort:

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

Überprüfen Sie den Systemstatus

> Wenn Sie OpenShift verwenden möchten, können Sie vergleichbare oc-Befehle für Verifizierungsschritte verwenden.

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

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

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

Beispielantwort:

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-5f75c5f564-bzqmt              1/1     Running   0
11m
activity-6b8f7cccb9-mlrn4                   1/1     Running   0
9m2s
api-token-authentication-6hznt              1/1     Running   0
8m50s
api-token-authentication-qpfgb              1/1     Running   0
8m50s
api-token-authentication-sqnb7              1/1     Running   0
8m50s
asup-5578bbdd57-dxkbp                       1/1     Running   0
9m3s
authentication-56bff4f95d-mspmq             1/1     Running   0
7m31s
bucketservice-6f7968b95d-9rrrl              1/1     Running   0
8m36s
cert-manager-5f6cf4bc4b-82khn               1/1     Running   0
6m19s
cert-manager-cainjector-76cf976458-sdrbc    1/1     Running   0
6m19s
cert-manager-webhook-5b7896bfd8-2n45j       1/1     Running   0
6m19s
cloud-extension-749d9f684c-8bdhq            1/1     Running   0
9m6s
cloud-insights-service-7d58687d9-h5tzw      1/1     Running   2
8m56s
composite-compute-968c79cb5-nv7l4           1/1     Running   0
9m11s
composite-volume-7687569985-jg9gg           1/1     Running   0
8m33s
```

```
credentials-5c9b75f4d6-nx9cz                1/1     Running   0
8m42s
entitlement-6c96fd8b78-zt7f8                1/1     Running   0
8m28s
features-5f7bfc9f68-gsjnl                   1/1     Running   0
8m57s
fluent-bit-ds-h88p7                         1/1     Running   0
7m22s
fluent-bit-ds-krhnj                         1/1     Running   0
7m23s
fluent-bit-ds-l5bjj                         1/1     Running   0
7m22s
fluent-bit-ds-lrclb                         1/1     Running   0
7m23s
fluent-bit-ds-s5t4n                         1/1     Running   0
7m23s
fluent-bit-ds-zpr6v                         1/1     Running   0
7m22s
graphql-server-5f5976f4bd-vbb4z             1/1     Running   0
7m13s
identity-56f78b8f9f-8h9p9                   1/1     Running   0
8m29s
influxdb2-0                                 1/1     Running   0
11m
krakend-6f8d995b4d-5khkl                    1/1     Running   0
7m7s
license-5b5db87c97-jmxzc                    1/1     Running   0
9m
login-ui-57b57c74b8-6xtv7                   1/1     Running   0
7m10s
loki-0                                      1/1     Running   0
11m
monitoring-operator-9dbc9c76d-8znck         2/2     Running   0
7m33s
nats-0                                      1/1     Running   0
11m
nats-1                                      1/1     Running   0
10m
nats-2                                      1/1     Running   0
10m
nautilus-6b9d88bc86-h8kfb                   1/1     Running   0
8m6s
nautilus-6b9d88bc86-vn68r                   1/1     Running   0
8m35s
openapi-b87d77dd8-5dz9h                     1/1     Running   0
9m7s
```

```
polaris-consul-consul-5ljfb                1/1     Running   0
11m
polaris-consul-consul-s5d5z                1/1     Running   0
11m
polaris-consul-consul-server-0             1/1     Running   0
11m
polaris-consul-consul-server-1             1/1     Running   0
11m
polaris-consul-consul-server-2             1/1     Running   0
11m
polaris-consul-consul-twmpq                1/1     Running   0
11m
polaris-mongodb-0                          2/2     Running   0
11m
polaris-mongodb-1                          2/2     Running   0
10m
polaris-mongodb-2                          2/2     Running   0
10m
polaris-ui-84dc87847f-zrg8w                1/1     Running   0
7m12s
polaris-vault-0                            1/1     Running   0
11m
polaris-vault-1                            1/1     Running   0
11m
polaris-vault-2                            1/1     Running   0
11m
public-metrics-657698b66f-67pgt            1/1     Running   0
8m47s
storage-backend-metrics-6848b9fd87-w7x8r   1/1     Running   0
8m39s
storage-provider-5ff5868cd5-r9hj7          1/1     Running   0
8m45s
telegraf-ds-dw4hg                          1/1     Running   0
7m23s
telegraf-ds-k92gn                          1/1     Running   0
7m23s
telegraf-ds-mmxjl                          1/1     Running   0
7m23s
telegraf-ds-nhs8s                          1/1     Running   0
7m23s
telegraf-ds-rj7lw                          1/1     Running   0
7m23s
telegraf-ds-tqrkb                          1/1     Running   0
7m23s
telegraf-rs-9mwgj                          1/1     Running   0
7m23s
```

```
telemetry-service-56c49d689b-ffrzx         1/1     Running   0
8m42s
tenancy-767c77fb9d-g9ctv                   1/1     Running   0
8m52s
traefik-5857d87f85-7pmx8                   1/1     Running   0
6m49s
traefik-5857d87f85-cpxgv                   1/1     Running   0
5m34s
traefik-5857d87f85-lvmlb                   1/1     Running   0
4m33s
traefik-5857d87f85-t2xlk                   1/1     Running   0
4m33s
traefik-5857d87f85-v9wpf                   1/1     Running   0
7m3s
trident-svc-595f84dd78-zb816               1/1     Running   0
8m54s
vault-controller-86c94fbf4f-krttq          1/1     Running   0
9m24s
```

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

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

> `accHost` Die Cluster-Registrierung ist einer der letzten Vorgänge, und bei Ausfall wird die Implementierung nicht fehlschlagen. Sollte ein Cluster-Registrierungsfehler in den Protokollen gemeldet werden, können Sie die Registrierung erneut durch den Add-Cluster-Workflow versuchen "In der UI" Oder API.

3. Wenn alle Pods ausgeführt werden, überprüfen Sie den Installationserfolg, indem Sie den abrufen `AstraControlCenter` Die Instanz wird vom Astra Control Center Operator installiert.

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

4. Prüfen Sie in der YAML die `status.deploymentState` Feld in der Antwort für das `Deployed` Wert: Wenn die Bereitstellung nicht erfolgreich war, wird stattdessen eine Fehlermeldung angezeigt.
5. Um das einmalige Passwort zu erhalten, das Sie bei der Anmeldung beim Astra Control Center verwenden, kopieren Sie das `status.uuid` Wert: Das Passwort lautet `ACC-` Anschließend der UUID-Wert (`ACC-[UUID]` Oder in diesem Beispiel `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Beispiele für YAML-Details

```
name: astra
   namespace: netapp-acc
   resourceVersion: "104424560"
   selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
   uid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
 spec:
   accountName: Example
   astraAddress: astra.example.com
   astraVersion: 21.12.60
   autoSupport:
     enrolled: true
     url: https://support.netapp.com/asupprod/post/1.0/postAsup
   crds: {}
   email: admin@example.com
   firstName: SRE
   imageRegistry:
     name: registry_name/astra
     secret: astra-registry-cred
   lastName: Admin
 status:
   accConditionHistory:
     items:
     - astraVersion: 21.12.60
       condition:
         lastTransitionTime: "2021-11-23T02:23:59Z"
         message: Deploying is currently in progress.
         reason: InProgress
         status: "False"
         type: Ready
       generation: 2
       observedSpec:
         accountName: Example
         astraAddress: astra.example.com
         astraVersion: 21.12.60
         autoSupport:
           enrolled: true
           url: https://support.netapp.com/asupprod/post/1.0/postAsup
         crds: {}
         email: admin@example.com
         firstName: SRE
         imageRegistry:
           name: registry_name/astra
           secret: astra-registry-cred
```

```
        lastName: Admin
      timestamp: "2021-11-23T02:23:59Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:23:59Z"
        message: Deploying is currently in progress.
        reason: InProgress
        status: "True"
        type: Deploying
      generation: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      timestamp: "2021-11-23T02:23:59Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Post Install was successful
        observedGeneration: 2
        reason: Complete
        status: "True"
        type: PostInstallComplete
      generation: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
```

```
          secret: astra-registry-cred
        lastName: Admin
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Deploying succeeded.
        reason: Complete
        status: "False"
        type: Deploying
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Astra is deployed
        reason: Complete
        status: "True"
        type: Deployed
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
```

```
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Astra is deployed
        reason: Complete
        status: "True"
        type: Ready
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
  certManager: deploy
  cluster:
    type: OCP
    vendorVersion: 4.7.5
    version: v1.20.0+bafe72f
  conditions:
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Astra is deployed
    reason: Complete
    status: "True"
    type: Ready
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Deploying succeeded.
    reason: Complete
```

```
      status: "False"
      type: Deploying
    - lastTransitionTime: "2021-12-08T16:19:53Z"
      message: Post Install was successful
      observedGeneration: 2
      reason: Complete
      status: "True"
      type: PostInstallComplete
    - lastTransitionTime: "2021-12-08T16:19:55Z"
      message: Astra is deployed
      reason: Complete
      status: "True"
      type: Deployed
    deploymentState: Deployed
    observedGeneration: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
    observedVersion: 21.12.60
    postInstall: Complete
    uuid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
kind: List
metadata:
 resourceVersion: ""
 selfLink: ""
```

Eindringen für den Lastenausgleich einrichten

Sie können einen Kubernetes Ingress-Controller einrichten, der den externen Zugriff auf Services, wie etwa den Lastausgleich in einem Cluster, managt.

Dieses Verfahren erklärt, wie ein Ingress-Controller eingerichtet wird (`ingressType:Generic`). Dies ist die Standardaktion mit Astra Control Center. Nachdem Astra Control Center bereitgestellt wurde, müssen Sie den Ingress-Controller so konfigurieren, dass Astra Control Center mit einer URL verfügbar ist.

> Wenn Sie keinen Ingress-Controller einrichten möchten, können Sie ihn einstellen `ingressType:AccTraefik)`. 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 nicht eine 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. Einzelheiten zum Servicetyp von „loadbalancer“ und Ingress finden Sie unter "Anforderungen".

Die Schritte unterscheiden sich je nach Art des Ingress-Controllers, den Sie verwenden:

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

Was Sie benötigen

- Erforderlich "Eingangs-Controller" Sollte bereits eingesetzt werden.
- Der "Eingangsklasse" Entsprechend der Eingangs-Steuerung sollte bereits erstellt werden.
- Sie verwenden Kubernetes-Versionen zwischen und v1.19 und v1.22.

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 mit entweder dem `v1beta1` (Veraltet in Kubernetes Version kleiner als oder 1.22) oder `v1` Ressourcentyp für ein deprecated oder ein neues Schema:
   a. Für A `v1beta1` Veraltete Schemas, folgen Sie diesem Beispiel:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: [class name for nginx controller]
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

b. Für das `v1` Neues Schema, folgen Sie diesem Beispiel:

```
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 addess>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

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, den Sie in verwendet haben `astraAddress` Im `astra_control_center_min.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_min.yaml` CR, wenn Sie haben das Astra Control Center 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".

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.

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
```

- Sie haben in Ihrem Rechenzentrum eine FQDN-Adresse für Astra Control Center erstellt.
- Sie verfügen über die erforderlichen Berechtigungen und haben Zugriff auf die Container-Plattform Red hat OpenShift, um die beschriebenen Installationsschritte durchzuführen.

Schritte

- Laden Sie das Astra Control Center Bundle herunter und entpacken Sie es aus
- 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 Bundle herunter und entpacken Sie es aus

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 aus dem 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
```

Installieren Sie das NetApp Astra kubectl Plug-in

Der NetApp Astra `kubectl` Kommandozeilen-Plugin spart Zeit bei der Ausführung von Routineaufgaben im Zusammenhang mit der Bereitstellung und dem Upgrade von Astra Control Center.

Was Sie benötigen

NetApp bietet Binärdateien für das Plug-in für verschiedene CPU-Architekturen und Betriebssysteme. Sie müssen wissen, welche CPU und welches Betriebssystem Sie haben, bevor Sie diese Aufgabe ausführen. Unter Linux- und Mac-Betriebssystemen können Sie die verwenden `uname -a` Befehl zum Sammeln dieser Informationen.

Schritte

1. Nennen Sie den verfügbaren NetApp Astra `kubectl` Plugin-Binärdateien, und notieren Sie den Namen der Datei, die Sie für Ihr Betriebssystem und CPU-Architektur benötigen:

```
ls kubectl-astra/
```

2. Kopieren Sie die Datei an denselben Speicherort wie der Standard `kubectl` Utility: In diesem Beispiel ist der `kubectl` Das Dienstprogramm befindet sich im `/usr/local/bin` Verzeichnis. Austausch `<binary-name>` Mit dem Namen der benötigten Datei:

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

Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

1. Wechseln Sie in das Astra-Verzeichnis:

```
cd acc
```

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

   Siehe Beispielskripts zum automatischen Laden von Bildern unten.

   a. Melden Sie sich bei Ihrer Registrierung an:

      Docker:

```
docker login [your_registry_path]
```

      Podman:

```
podman login [your_registry_path]
```

   b. Verwenden Sie das entsprechende Skript, um die Bilder zu laden, die Bilder zu kennzeichnen, und Schieben Sie die Bilder in Ihre lokale Registrierung:

      Docker:

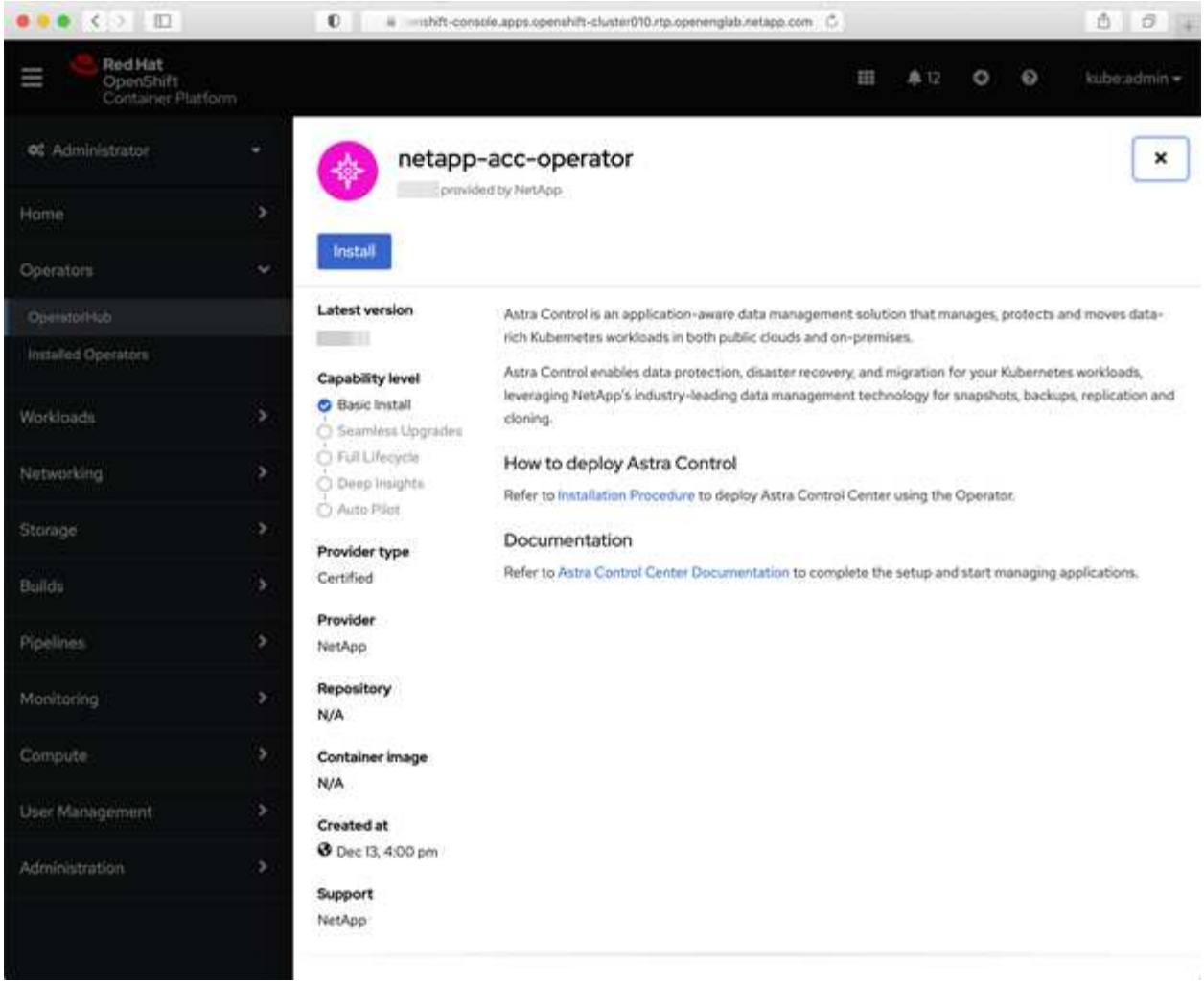
```
export REGISTRY=[Docker_registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image
trimming the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
image: //')
  astraImage=$(echo ${astraImage} |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

Podman:

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

Suchen Sie die Installationsseite des Bedieners

1. Führen Sie eines der folgenden Verfahren aus, um auf die Installationsseite des Bedieners zuzugreifen:
   - Von der Red hat OpenShift-Webkonsole
     aus:



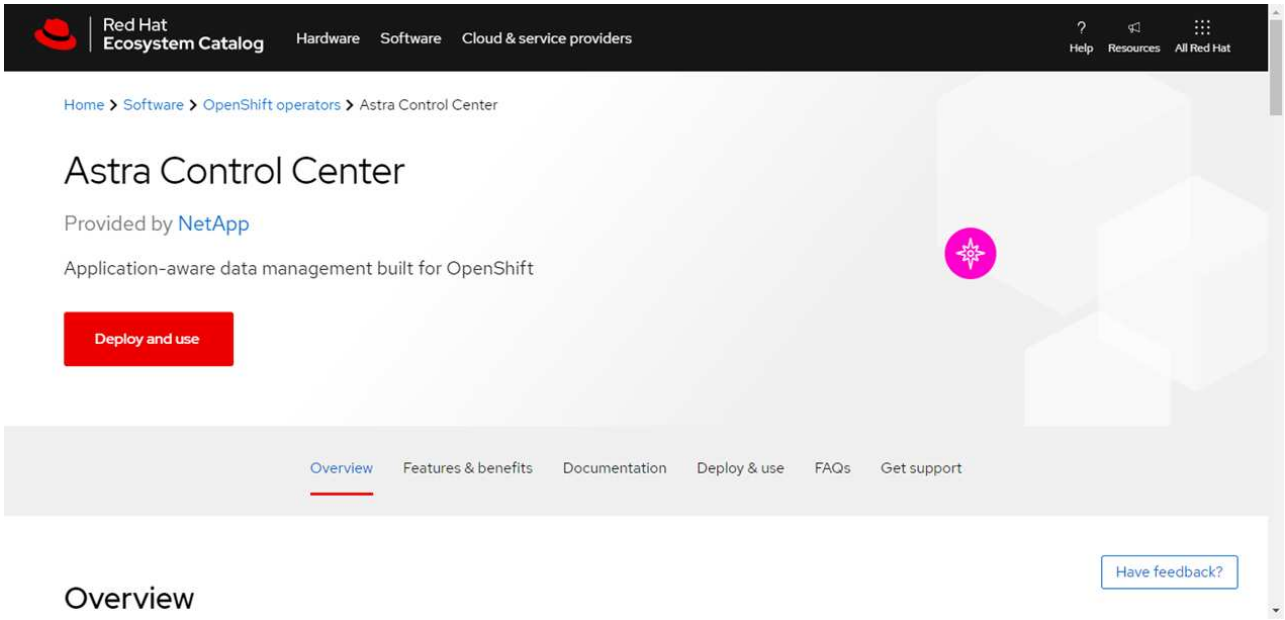
i. Melden Sie sich in der OpenShift Container Platform UI an.

ii. Wählen Sie im Seitenmenü die Option **Operatoren > OperatorHub** aus.

iii. Wählen Sie den Operator des NetApp Astra Control Center aus.

iv. Wählen Sie **Installieren**.

- 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 in der Detailansicht des Bedieners Astra Control Center die Option aus `Create instance` Im Abschnitt über die bereitgestellten APIs.

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

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

   b. (Optional) Aktivieren oder Deaktivieren von Auto Support. Es wird empfohlen, die Auto Support-

Funktion beizubehalten.

c. Geben Sie die Astra Control Center-Adresse ein. Kommen Sie nicht herein `http://` Oder `https://` In der Adresse.

d. Geben Sie die Astra Control Center-Version ein, z. B. 21.12.60.

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

f. Beibehaltung der Standard-Richtlinie zur Rückgewinnung von Volumes

g. Geben Sie in **Image Registry** Ihren lokalen Container Image Registry-Pfad ein. Kommen Sie nicht herein `http://` Oder `https://` In der Adresse.

h. Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, geben Sie das Geheimnis ein.

i. Geben Sie den Vornamen des Administrators ein.

j. Konfiguration der Ressourcenskalierung

k. Behalten Sie die Standard-Storage-Klasse bei.

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

3. Wählen Sie `Create`.

Wie es weiter geht

Überprüfen Sie die erfolgreiche Installation von Astra Control Center und führen Sie die "Verbleibende Schritte" Um sich anzumelden. Darüber hinaus wird die Implementierung abgeschlossen, indem Sie auch die Ausführung durchführen "Setup-Aufgaben".

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.

Um Astra Control Center in Amazon Web Services (AWS) und Microsoft Azure mit einem Cloud Volumes ONTAP Storage-Backend zu installieren, 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 Microsoft Azure

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.

Nur OCP-Cluster (Self-Managed OpenShift Container Platform) werden für die Bereitstellung von Astra Control Center unterstützt.

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
- OCP-Berechtigungen (Container Platform) für Red hat OpenShift (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 ist die gehostete AWS Zone und der Eintrag Route 53 erforderlich

Anforderungen der Betriebsumgebung für AWS

Astra Control Center erfordert die folgende Betriebsumgebung für AWS:

- Red hat OpenShift Container Platform 4.8

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

Das Astra Control Center benötigt 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
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 als Teil der Kubernetes Cluster Discovery in NetApp Cloud Manager)	Astra Trident 21.04 oder höher ist installiert und konfiguriert und NetApp ONTAP Version 9.5 oder höher als Storage-Backend

Komponente	Anforderungen
Bildregistrierung	Sie müssen über eine vorhandene private Registry, wie AWS Elastic Container Registry, mit der Sie Astra Control Center Build-Images 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 ONTAP Kubernetes Storage-Klassen, die beim Importieren des Kubernetes Clusters in NetApp 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. NetApp Cloud Manager konfigurieren.

5. Installieren Sie Astra Control Center.

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

Stellen Sie sicher, dass die IAM-Rollen und -Berechtigungen ausreichend sind, damit ein RedHat OpenShift-Cluster und ein NetApp Cloud Manager Connector installiert werden 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 dann AWS für die Erstellung eines virtuellen Netzwerks, richten Sie EC2 Computing-Instanzen ein, erstellen Sie einen AWS S3-Bucket, erstellen Sie ein Elastic Container Register (ECR), um die Astra Control Center Images zu hosten und übertragen Sie die Images auf diese Registrierung.

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

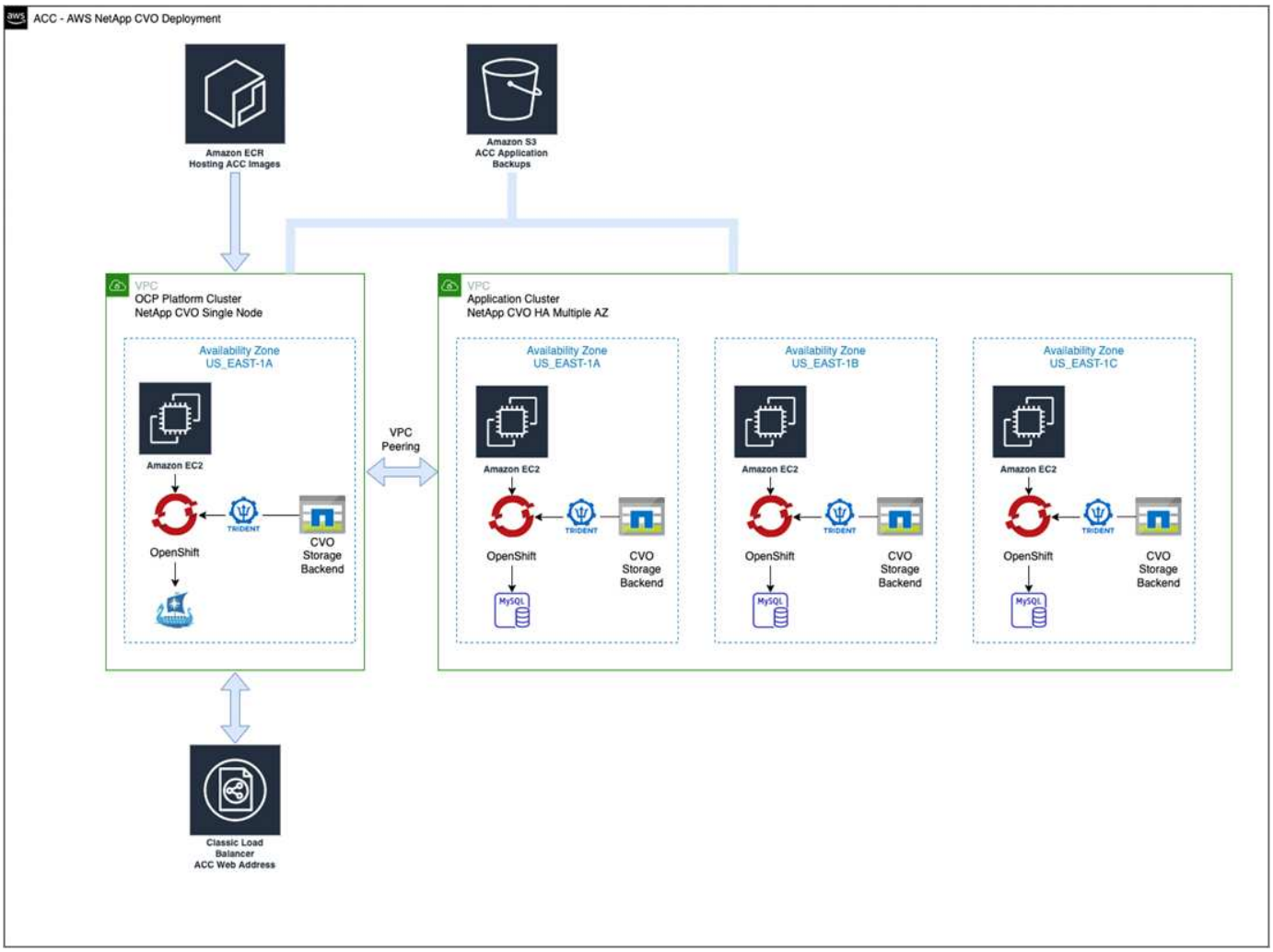
1. Virtuelles AWS Netzwerk erstellen.
2. EC2 Computing-Instanzen prüfen. Dabei können es sich um einen Bare Metal Server oder VMs in AWS handelt.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in 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. AWS Elastic Container Registry (ECR) erstellen, um alle ACC-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.

6. Drücken Sie die ACC-Bilder auf die 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:



NetApp Cloud Manager konfigurieren

Erstellen Sie mit Cloud Manager einen Workspace, fügen Sie eine Connector zu AWS hinzu, erstellen Sie eine Arbeitsumgebung und importieren Sie den Cluster.

Folgen Sie der Dokumentation zum Cloud Manager, um die folgenden Schritte auszuführen. Siehe folgendes:

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

Schritte

1. Fügen Sie Ihre Zugangsdaten zu Cloud Manager 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 oben rechts die 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. Trident wird automatisch im Rahmen des Import- und Erkennungsvorgangs 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

Dem Standard folgen "Installationsanweisungen für Astra Control Center".

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
- Red hat OpenShift Container Platform (OCP) 4.8
- OCP-Berechtigungen (Container Platform) für Red hat OpenShift (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 Anforderungen an die Ressourcen entspricht, die in der offiziellen Dokumentation der Umgebung aufgeführt sind.

Das Astra Control Center benötigt 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

Komponente	Anforderungen
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 als Teil der Kubernetes Cluster Discovery in NetApp Cloud Manager)	Astra Trident 21.04 oder neuer installiert und konfiguriert und NetApp ONTAP Version 9.5 oder neuer wird als Storage-Backend verwendet
Bildregistrierung	Sie müssen über eine vorhandene private Registry, wie z. B. Azure Container Registry (ACR) verfügen, in die Sie Bilder vom Astra Control Center erstellen 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 Importieren des Kubernetes Clusters in NetApp 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.

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. NetApp Cloud Manager konfigurieren.
6. Installation und Konfiguration des Astra Control Center.

Installieren Sie einen RedHat OpenShift-Cluster auf Azure

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

Installationsanweisungen finden Sie in der Dokumentation zu RedHat auf "Installation von OpenShift-Cluster auf Azure" Und "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 die IAM-Rollen und -Berechtigungen ausreichend sind, damit ein RedHat OpenShift-Cluster und ein NetApp Cloud Manager Connector installiert werden können.

Siehe "Azure Zugangsdaten und Berechtigungen".

Konfigurieren Sie Azure

Konfigurieren Sie dann Azure für die Erstellung eines virtuellen Netzwerks, richten Sie Computing-Instanzen ein, erstellen Sie einen Azure Blob Container, erstellen Sie ein Azure Container Register (ACR), um die Astra Control Center Images zu hosten und übertragen Sie die Bilder auf diese Registrierung.

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. Erstellen Sie eine Azure Container Registry (ACR), um alle Astra Control Center-Images zu hosten.
8. ACR-Zugriff für Docker-Push/Pull-alle Astra Control Center-Images einrichten.
9. Drücken Sie die ACC-Bilder in diese Registrierung, indem Sie das folgende Skript eingeben:

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

   - Beispiel*:

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

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 $AZ_ACR_REGISTRYY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

10. Richten Sie DNS-Zonen ein.

NetApp Cloud Manager konfigurieren

Erstellen Sie mit Cloud Manager einen Workspace, fügen Sie einen Connector zu Azure hinzu, erstellen Sie eine Arbeitsumgebung und importieren Sie den Cluster.

Folgen Sie der Dokumentation zum Cloud Manager, um die folgenden Schritte auszuführen. Siehe "Erste Schritte mit Cloud Manager in Azure".

Was Sie benötigen

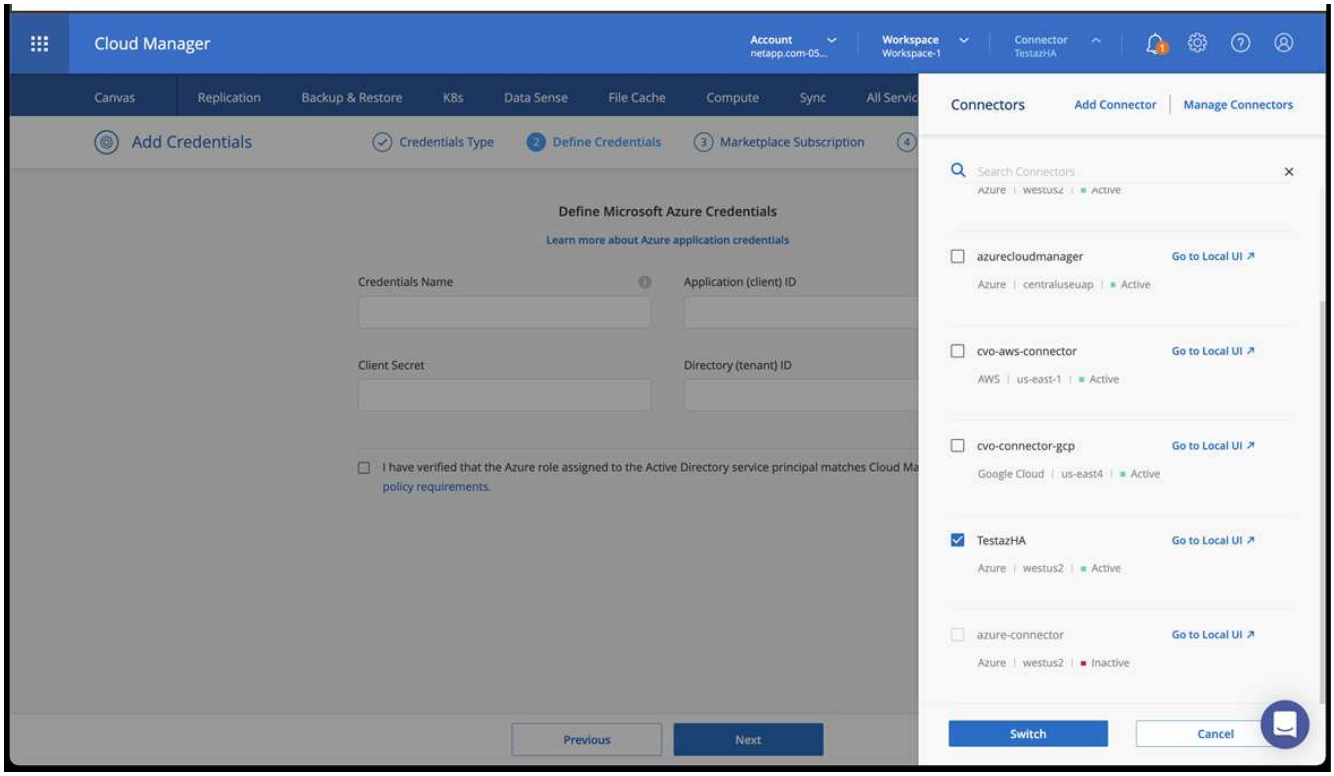
Zugriff auf das Azure Konto mit den erforderlichen IAM-Berechtigungen und -Rollen

Schritte

1. Fügen Sie Ihre Zugangsdaten zu Cloud Manager hinzu.
2. Fügen Sie einen Connector für Azure hinzu. Siehe "Richtlinien für Cloud Manager".
   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 Cloud Manager".

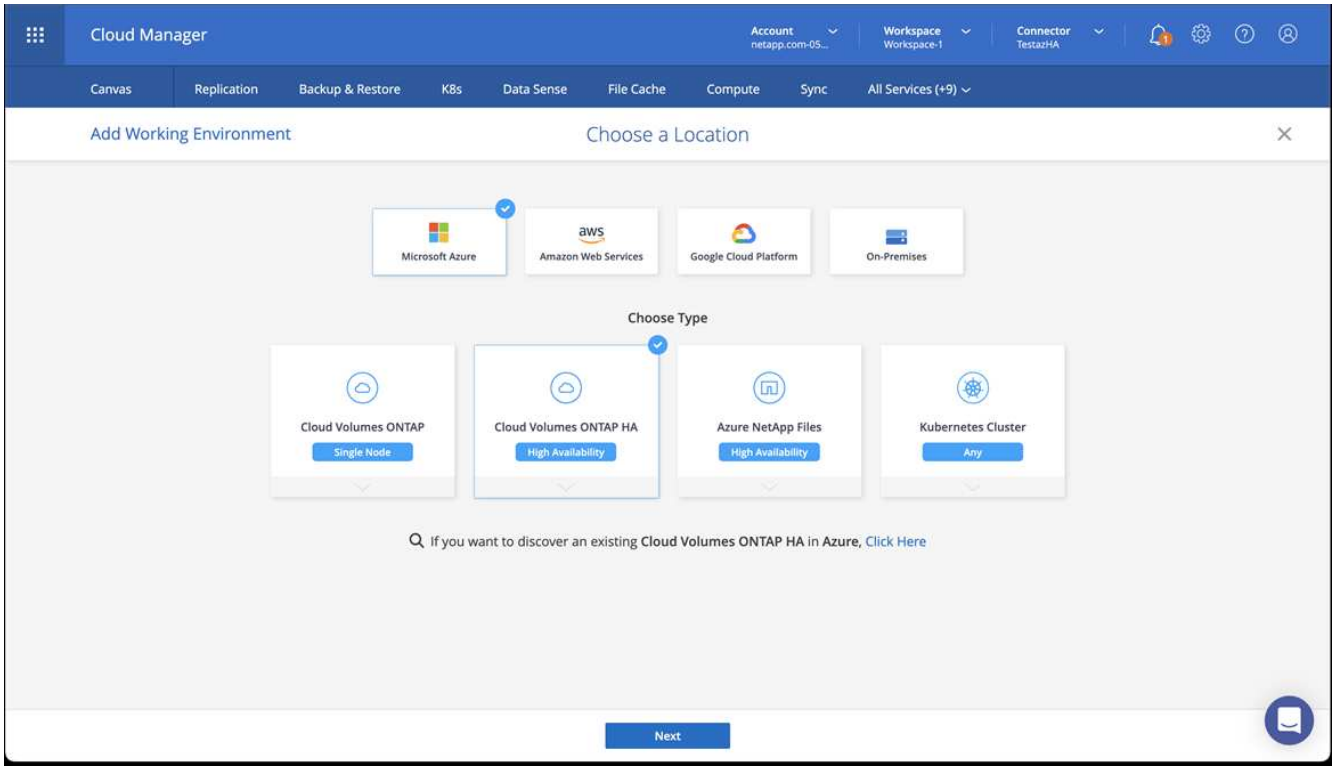
3. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.



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

   a. Ort: „Microsoft Azure“.

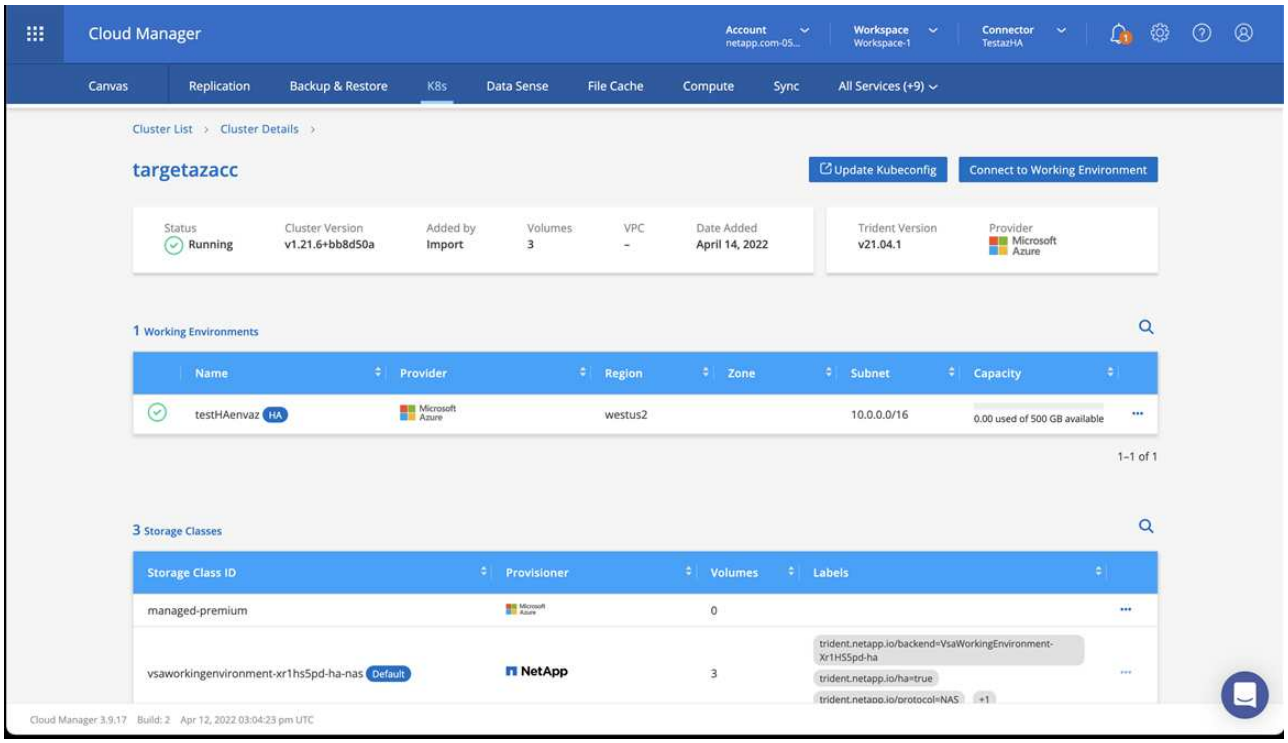
   b. Typ: „Cloud Volumes ONTAP HA“.



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 oben rechts die 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. Trident wird automatisch im Rahmen des Import- und Erkennungsvorgangs 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 des Astra Control Center

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".

Copyright-Informationen

Copyright © 2023 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.