Einrichten des Astra Control Center

Astra Control Center

NetApp
April 25, 2024

Inhalt

Einrichten des Astra Control Center . . . 1
Fügen Sie eine Lizenz für Astra Control Center hinzu . . . 1
Astra Control Provisioner Aktivieren . . . 1
Bereiten Sie Ihre Umgebung mit Astra Control auf das Cluster-Management vor . . . 12
(Tech Preview) Installieren Sie Astra Connector für gemanagte Cluster . . . 24
Fügen Sie einen Cluster hinzu . . . 27
Aktivieren Sie die Authentifizierung auf einem ONTAP Storage Back-End . . . 28
Fügen Sie ein Storage-Back-End hinzu . . . 35
Fügen Sie einen Bucket hinzu . . . 36

Einrichten des Astra Control Center

Fügen Sie eine Lizenz für Astra Control Center hinzu

Wenn Sie Astra Control Center installieren, ist bereits eine eingebettete Evaluierungslizenz installiert. Wenn Sie Astra Control Center evaluieren, können Sie diesen Schritt überspringen.

Über die Astra Control UI oder können Sie eine neue Lizenz hinzufügen "Astra Control API".

Astra Control Center Lizenzen messen die CPU-Ressourcen mithilfe von Kubernetes-CPU-Einheiten und berücksichtigen die CPU-Ressourcen, die den Worker-Nodes aller gemanagten Kubernetes-Cluster zugewiesen sind. Lizenzen basieren auf der vCPU-Nutzung. Weitere Informationen zur Berechnung von Lizenzen finden Sie unter "Lizenzierung".

> 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.

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

Bevor Sie beginnen

- Zugriff auf eine neu installierte Astra Control Center-Instanz.
- Berechtigungen für Administratorrollen.
- A "NetApp Lizenzdatei" (NLF).

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 und keine Daten an AutoSupport senden, sollten Sie Ihre Konto-ID speichern, um Datenverlust im Falle eines Astra Control Center-Ausfalls zu vermeiden.

Astra Control Provisioner Aktivieren

In Astra Trident Version 23.10 und höher können Sie Astra Control Provisioner verwenden, damit lizenzierte Benutzer von Astra Control auf erweiterte Storage-Bereitstellungsfunktionen zugreifen können. Astra Control Provisioner bietet diese

erweiterte Funktionalität zusätzlich zu den auf Astra Trident basierenden Standardfunktionen.

Bei den neuesten Updates für Astra Control wird Astra Control Provisioner Astra Trident als Storage-bereitstellung und -Orchestrierung ersetzen und für die Verwendung von Astra Control obligatorisch sein. Aus diesem Grund wird dringend empfohlen, Astra Control für die Astra Control-Bereitstellung zu aktivieren. Astra Trident wird weiterhin Open Source bleiben und mit neuen CSI- und anderen Funktionen von NetApp veröffentlicht, gepflegt, unterstützt und auf dem neuesten Stand sein.

Über diese Aufgabe

Befolgen Sie dieses Verfahren, wenn Sie als lizenzierter Astra Control Center-Benutzer die Astra Control Provisioner-Funktion verwenden möchten. Wenn Sie Astra Trident verwenden und die zusätzlichen Funktionen von Astra Control Provisioner verwenden möchten, sollten Sie dieses Verfahren auch befolgen.

In allen Fällen ist die bereitstellungsfunktion in Astra Trident 24.02 standardmäßig nicht aktiviert und muss aktiviert sein.

Bevor Sie beginnen

Wenn Sie die Astra Control Provisioner aktivieren, gehen Sie wie folgt vor:

Astra Control Provisioniert Benutzer mit Astra Control Center

- **Bewirke eine Astra Control Center Lizenz**: Du benötigst eine "Astra Control Center-Lizenz" Um Astra Control Provisioner zu aktivieren und auf die enthaltenen Funktionen zuzugreifen.
- **Installation oder Upgrade auf Astra Control Center 23.10 oder höher**: Sie benötigen die neueste Version von Astra Control Center (24.02), wenn Sie die neueste Astra Control Provisioner-Funktion (24.02) mit Astra Control verwenden möchten.
- **Bestätigen Sie, dass Ihr Cluster über eine AMD64-Systemarchitektur verfügt**: Das Astra Control Provisioner-Image wird sowohl in AMD64- als auch in ARM64-CPU-Architekturen bereitgestellt, aber nur AMD64 wird von Astra Control Center unterstützt.
- **Erhalten Sie ein Astra Control Service-Konto für den Registrierungszugriff**: Wenn Sie beabsichtigen, die Astra Control-Registrierung anstelle der NetApp-Support-Website zu verwenden, um das Astra Control-Provisioner-Image herunterzuladen, schließen Sie die Registrierung für einen ab "Astra Control Service Konto". Nachdem Sie das Formular ausgefüllt, übermittelt und ein BlueXP Konto erstellt haben, erhalten Sie eine Willkommens-E-Mail für Astra Control Service.
- **Wenn Sie Astra Trident installiert haben, bestätigen Sie, dass seine Version innerhalb eines Fensters mit vier Versionen ist**: Sie können ein direktes Upgrade auf Astra Trident 24.02 mit Astra Control Provisioner durchführen, wenn Ihr Astra Trident innerhalb eines Fensters mit vier Versionen von Version 24.02 ist. Sie können beispielsweise direkt von Astra Trident 23.04 auf 24.02 aktualisieren.

Astra Control Provisioner nur Benutzer

- **Bewirke eine Astra Control Center Lizenz**: Du benötigst eine "Astra Control Center-Lizenz" Um Astra Control Provisioner zu aktivieren und auf die enthaltenen Funktionen zuzugreifen.
- **Wenn Sie Astra Trident installiert haben, bestätigen Sie, dass seine Version innerhalb eines Fensters mit vier Versionen ist**: Sie können ein direktes Upgrade auf Astra Trident 24.02 mit Astra Control Provisioner durchführen, wenn Ihr Astra Trident innerhalb eines Fensters mit vier Versionen von Version 24.02 ist. Sie können beispielsweise direkt von Astra Trident 23.04 auf 24.02 aktualisieren.
- **Sie können ein Astra Control Service-Konto für den Registrierungszugriff abrufen**: Sie benötigen Zugriff auf die Registrierung, um Astra Control Provisioner-Bilder herunterladen zu können. Um zu beginnen, füllen Sie die Registrierung für einen aus "Astra Control Service Konto". Nachdem Sie das Formular ausgefüllt, übermittelt und ein BlueXP Konto erstellt haben, erhalten Sie eine Willkommens-E-Mail für Astra Control Service.

(Schritt 1) Holen Sie sich das Bild zur Astra Control Bereitstellung

Benutzer von Astra Control Center können das Image für die Astra Control Provisioner entweder über die Astra Control Registry oder die NetApp Support Site erhalten. Für Astra Trident Benutzer, die Astra Control Provisioner ohne Astra Control verwenden möchten, sollte die Registrierungsmethode verwendet werden.

Astra Control-Image-Registrierung

> Bei diesem Verfahren können Sie Podman anstelle von Docker für die Befehle verwenden. Wenn Sie eine Windows-Umgebung verwenden, wird PowerShell empfohlen.

1. Rufen Sie die NetApp Astra Control Image-Registry auf:
   a. Melden Sie sich bei der Astra Control Service Web-UI an, und wählen Sie das Symbol oben rechts auf der Seite aus.
   b. Wählen Sie **API-Zugriff**.
   c. Notieren Sie sich Ihre Konto-ID.
   d. Wählen Sie auf derselben Seite **API-Token generieren** aus und kopieren Sie die API-Token-Zeichenfolge in die Zwischenablage und speichern Sie sie in Ihrem Editor.
   e. Melden Sie sich über Ihre bevorzugte Methode in der Astra Control Registry an:

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

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

2. (Nur benutzerdefinierte Registrierungen) Befolgen Sie diese Schritte, um das Bild in Ihre benutzerdefinierte Registrierung zu verschieben. Wenn Sie keine Registrierung verwenden, befolgen Sie die Schritte des Trident-Operators in "Nächster Abschnitt".
   a. Rufen Sie das Astra Control Provisioner-Image aus der Registrierung ab:

> Das abgezogene Image unterstützt nicht mehrere Plattformen und unterstützt nur die gleiche Plattform wie der Host, der das Image gezogen hat, wie z. B. Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Beispiel:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform
linux/amd64
```

   a. Markieren Sie das Bild:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

b. Laden Sie das Bild in Ihre benutzerdefinierte Registrierung:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

> Alternativ zum Ausführen der folgenden Docker Befehle können Sie Crane Copy verwenden:
> `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

NetApp Support Website

1. Laden Sie das Bundle für die Astra Control Bereitstellung herunter (`trident-acp-[version].tar`) Vom "Download-Seite für Astra Control Center".
2. (Empfohlen, aber optional) Laden Sie das Paket Zertifikate und Signaturen für Astra Control Center (astra-control-Center-certs-[Version].tar.gz) herunter, um die Signatur des tar-Bundles tar von trident-acp-[Version] zu überprüfen.

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

```
openssl dgst -sha256 -verify certs/AstraControlCenterDockerImages-public.pub -signature certs/trident-acp-[version].tar.sig trident-acp-[version].tar
```

3. Laden Sie das Astra Control Provisioner-Bild:

```
docker load < trident-acp-24.02.0.tar
```

Antwort:

```
Loaded image: trident-acp:24.02.0-linux-amd64
```

4. Markieren Sie das Bild:

```
docker tag trident-acp:24.02.0-linux-amd64 <my_custom_registry>/trident-acp:24.02.0
```

5. Laden Sie das Bild in Ihre benutzerdefinierte Registrierung:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

(Schritt 2) Aktivieren Sie die Astra Control-Bereitstellung in Astra Trident

Stellen Sie fest, ob die ursprüngliche Installationsmethode einen verwendet hat "Operator (manuell oder mit Helm) oder tridentctl" Und führen Sie die entsprechenden Schritte entsprechend Ihrer ursprünglichen Methode durch.

Astra Trident Betreiber

1. "Laden Sie das Astra Trident Installationsprogramm herunter und extrahieren Sie es".
2. Führen Sie diese Schritte aus, wenn Sie Astra Trident noch nicht installiert haben oder den Operator aus der ursprünglichen Astra Trident-Implementierung entfernt haben:
   a. Erstellen des CRD:

   ```
   kubectl create -f
   deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.y
   aml
   ```

   b. Erstellen Sie den Namespace für Trident (`kubectl create namespace trident`) Oder bestätigen Sie, dass der Namensraum Dreizack noch existiert (`kubectl get all -n trident`). Wenn der Namespace entfernt wurde, erstellen Sie ihn erneut.
3. Update von Astra Trident auf 24.02.0:

   > Verwenden Sie für Cluster mit Kubernetes 1.24 oder früheren Versionen `bundle_pre_1_25.yaml`. Verwenden Sie für Cluster mit Kubernetes 1.25 oder höher `bundle_post_1_25.yaml`.

   ```
   kubectl -n trident apply -f trident-installer/deploy/<bundle-
   name.yaml>
   ```

4. Überprüfen Sie, ob Astra Trident ausgeführt wird:

   ```
   kubectl get torc -n trident
   ```

   Antwort:

   ```
   NAME       AGE
   trident    21m
   ```

5. Wenn Sie eine Registry mit Geheimnissen haben, erstellen Sie ein Geheimnis, mit dem Sie das Astra Control Provisioner-Bild abrufen können:

   ```
   kubectl create secret docker-registry <secret_name> -n trident
   --docker-server=<my_custom_registry> --docker-username=<username>
   --docker-password=<token>
   ```

6. Bearbeiten Sie den TridentOrchestrator CR, und nehmen Sie die folgenden Änderungen vor:

```
kubectl edit torc trident -n trident
```

a. Legen Sie einen benutzerdefinierten Registrierungsport für das Astra Trident Image fest oder ziehen Sie es aus der Astra Control Registry (`tridentImage: <my_custom_registry>/trident:24.02.0` Oder `tridentImage: netapp/trident:24.02.0`).

b. Astra Control Provisioner Aktivieren (`enableACP: true`).

c. Legen Sie den benutzerdefinierten Registrierungsport für das Astra Control Provisioner-Image fest oder ziehen Sie es aus der Astra Control Registry (`acpImage: <my_custom_registry>/trident-acp:24.02.0` Oder `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Wenn Sie sich etabliert haben Geheimnisse der Bildausziehung Sie können diese hier einstellen (`imagePullSecrets: - <secret_name>`). Verwenden Sie den gleichen geheimen Namen, den Sie in den vorherigen Schritten festgelegt haben.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
  - <secret_name>
```

7. Speichern und beenden Sie die Datei. Der Bereitstellungsprozess wird automatisch gestartet.
8. Überprüfen Sie, ob der Operator, die Bereitstellung und Replikasets erstellt wurden.

```
kubectl get all -n trident
```

> Es sollte nur eine Instanz* des Operators in einem Kubernetes-Cluster geben. Erstellen Sie nicht mehrere Implementierungen des Astra Trident Operators.

9. Überprüfen Sie die `trident-acp` Container läuft und das `acpVersion` Ist `24.02.0` Mit dem Status `Installed`:

```
kubectl get torc -o yaml
```

Antwort:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

Tridentctl

1. "Laden Sie das Astra Trident Installationsprogramm herunter und extrahieren Sie es".
2. "Wenn Sie bereits Astra Trident verwenden, deinstallieren Sie ihn aus dem Cluster, das ihn hostet".
3. Installieren Sie Astra Trident mit aktiviertem Astra Control Provisioner (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Aktivieren Sie die Astra Control Provisioner-Funktion:

```
./tridentctl -n trident version
```

Antwort:

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 24.02.0 | 24.02.0 | 24.02.0. | +----------------
+----------------+-------------+
```

Helm

1. Bei Astra Trident 23.07.1 oder einer früheren Version "Deinstallieren" Der Bediener und andere Komponenten.
2. Wenn auf dem Kubernetes-Cluster 1.24 oder eine frühere Version ausgeführt wird, löschen Sie psp:

```
kubectl delete psp tridentoperatorpod
```

3. Fügen Sie das Helm Repository von Astra Trident hinzu:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

4. Aktualisieren Sie das Helm-Diagramm:

```
helm repo update netapp-trident
```

Antwort:

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. □Happy Helming!□
```

5. Auflisten der Bilder:

```
./tridentctl images -n trident
```

Antwort:

```
| v1.28.0            | netapp/trident:24.02.0|
|                    | docker.io/netapp/trident-autosupport:24.02|
|                    | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                    | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                    | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                    | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                    | registry.k8s.io/sig-storage/csi-node-driver-
registrar:v2.10.0 |
|                    | netapp/trident-operator:24.02.0 (optional)
```

6. Stellen Sie sicher, dass Dreizack-Bediener 24.02.0 verfügbar ist:

```
helm search repo netapp-trident/trident-operator --versions
```

Antwort:

```
NAME                            CHART VERSION   APP VERSION
DESCRIPTION
netapp-trident/trident-operator 100.2402.0      24.02.0         A
```

7. Nutzung `helm install` Und führen Sie eine der folgenden Optionen aus, die diese Einstellungen enthalten:
   - Ein Name für Ihren Bereitstellungsort
   - Die Version Astra Trident
   - Der Name des Bildes für die Astra Control-Bereitstellung
   - Das Flag, mit dem die provisionierung aktiviert wird
   - (Optional) Ein lokaler Registrierungspfad. Wenn Sie eine lokale Registrierung verwenden, wird Ihr "Trident Images" Kann in einer Registrierung oder in verschiedenen Registern gefunden werden, aber alle CSI-Images müssen sich in derselben Registrierung befinden.
   - Der Trident Namespace

Optionen

- Bilder ohne Registrierung

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0
--set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Bilder in einer oder mehreren Registern

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

Verwenden Sie können `helm list` So prüfen Sie Installationsdetails wie Name, Namespace, Diagramm, Status, App-Version, Und Revisionsnummer.

Falls Sie Probleme bei der Implementierung von Trident mit Helm haben, führen Sie diesen Befehl aus, um Astra Trident vollständig zu deinstallieren:

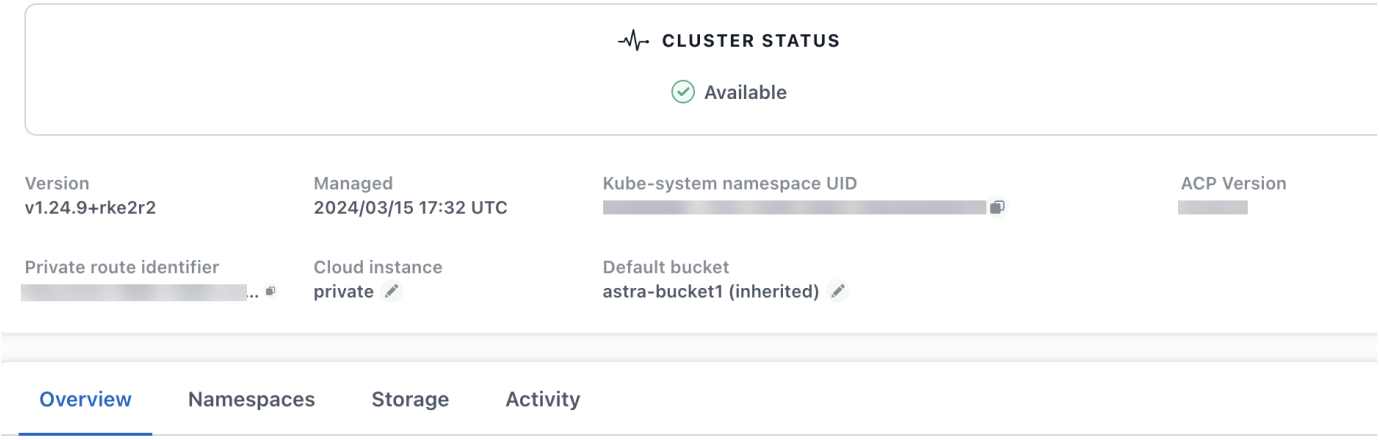
```
./tridentctl uninstall -n trident
```

Nicht "Astra Trident CRDs vollständig entfernen" Im Rahmen der Deinstallation vor dem erneuten Versuch, Astra Control Provisioner zu aktivieren.

Ergebnis

Die Bereitstellungsfunktion von Astra Control ist aktiviert und Sie können alle Funktionen der verwendeten Version verwenden.

(Nur für Astra Control Center Benutzer) nach der Installation von Astra Control wird für das Cluster, das die provisionierung in der Astra Control Center UI hostet, ein angezeigt `ACP version` Und nicht `Trident version` Feld und aktuelle installierte Versionsnummer.



Finden Sie weitere Informationen

- "Dokumentation für Astra Trident Upgrades"

Bereiten Sie Ihre Umgebung mit Astra Control auf das Cluster-Management vor

Sie sollten sicherstellen, dass die folgenden Voraussetzungen erfüllt sind, bevor Sie ein Cluster hinzufügen. Außerdem sollten Sie Eignungsprüfungen durchführen, um sicherzustellen, dass Ihr Cluster zum Astra Control Center hinzugefügt werden kann, und bei Bedarf kubeconfig-Clusterrollen erstellen.

Mit Astra Control können Sie je nach Umgebung und Einstellungen Cluster hinzufügen, die von Custom Resource (CR) oder kubeconfig gemanagt werden.

Bevor Sie beginnen

- **Umweltvoraussetzungen erfüllen**: Ihre Umgebung erfüllt sich "Anforderungen an die Betriebsumgebung" Für Astra Control Center.
- **Konfigurieren Sie Worker-Knoten**: Stellen Sie sicher, dass Sie "Konfigurieren Sie die Worker-Knoten" In Ihrem Cluster mit den entsprechenden Storage-Treibern erstellen, damit die Pods mit dem Back-End Storage interagieren können.
- **PSA-Einschränkungen aktivieren**: Wenn in Ihrem Cluster die Durchsetzung von Pod-Sicherheitszulassung aktiviert ist, was standardmäßig für Cluster ab Kubernetes 1.25 gilt, müssen Sie die PSA-Einschränkungen für diese Namespaces aktivieren:

- `netapp-acc-operator` Namespace:

```
kubectl label --overwrite ns netapp-acc-operator pod-
security.kubernetes.io/enforce=privileged
```

- `netapp monitoring` Namespace:

```
kubectl label --overwrite ns netapp-monitoring pod-
security.kubernetes.io/enforce=privileged
```

- **ONTAP-Anmeldeinformationen**: Sie benötigen ONTAP-Anmeldeinformationen und eine Superuser- und Benutzer-ID auf dem Backing-ONTAP-System, um Apps mit Astra Control Center zu sichern und wiederherzustellen.

  Führen Sie die folgenden Befehle in der ONTAP-Befehlszeile aus:

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **Kubeconfig-Managed Cluster Requirements**: Diese Anforderungen sind spezifisch für App-Cluster, die von kubeconfig verwaltet werden.
  - **Kubeconfig zugänglich machen**: Sie haben Zugang zum "Standardcluster kubeconfig" Das "Sie haben während der Installation konfiguriert".
  - **Hinweise zur Zertifizierungsstelle**: Wenn Sie den Cluster mit einer kubeconfig-Datei hinzufügen, die auf eine private Zertifizierungsstelle verweist, fügen Sie die folgende Zeile zum hinzu `cluster` Abschnitt der Datei kubeconfig. So kann Astra Control das Cluster hinzufügen:

```
insecure-skip-tls-verify: true
```

  - **Rancher only**: Ändern Sie beim Verwalten von Anwendungsclustern in einer Rancher-Umgebung den Standardkontext des Anwendungsclusters in der von Rancher bereitgestellten kubeconfig-Datei, um einen Steuerebenen-Kontext anstelle des Rancher API-Serverkontexts zu verwenden. So wird die Last auf dem Rancher API Server reduziert und die Performance verbessert.
- **Anforderungen für die Astra Control-Bereitstellung**: Sie sollten einen ordnungsgemäß konfigurierten Astra Control Provisioner einschließlich der Astra Trident-Komponenten verwenden, um Cluster zu managen.
  - **Umgebungsanforderungen für Astra Trident prüfen**: Lesen Sie vor der Installation oder dem Upgrade von Astra Control Provisioner die "Unterstützte Frontends, Back-Ends und Host-Konfigurationen".
  - **Astra Control-Provisioner aktivieren**: Es wird dringend empfohlen, Astra Trident 23.10 oder höher zu installieren und zu aktivieren "Astra Control bietet erweiterte Storage-Funktionen zur Bereitstellung". In den kommenden Versionen unterstützt Astra Control nicht Astra Trident, wenn der Astra Control

Provisioner nicht ebenfalls aktiviert ist.

- **Konfiguration eines Speicher-Backends**: Mindestens ein Speicher-Backend muss sein "In Astra Trident konfiguriert" Auf dem Cluster.
- **Konfiguration einer Storage-Klasse**: Mindestens eine Storage-Klasse muss sein "In Astra Trident konfiguriert" Auf dem Cluster. Wenn eine Standardspeicherklasse konfiguriert ist, stellen Sie sicher, dass sie die **einzige** Speicherklasse ist, die die Standardanmerkung hat.
- **Konfigurieren Sie einen Volume-Snapshot-Controller und installieren Sie eine Volume-Snapshot-Klasse**: "Installieren Sie einen Volume-Snapshot-Controller" Damit Snapshots in Astra Control erstellt werden können. "Erstellen" Mindestens eine `VolumeSnapshotClass` Einsatz von Astra Trident:

Führen Sie Eignungsprüfungen durch

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

Schritte

1. Bestimmen Sie die Astra Trident-Version, die Sie ausführen:

   ```
   kubectl get tridentversion -n trident
   ```

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

   ```
   NAME      VERSION
   trident   24.02.0
   ```

   Wenn Astra Trident nicht existiert, wird eine Ausgabe wie die folgende angezeigt:

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

2. Führen Sie einen der folgenden Schritte aus:
   - Wenn Sie Astra Trident 23.01 oder eine frühere Version verwenden, verwenden Sie diese "Anweisungen" Das Upgrade auf eine neuere Version von Astra Trident erfolgt vor dem Upgrade auf Astra Control Provisioner. Das können Sie "Führen Sie ein direktes Upgrade durch" Astra Control Provisioner 24.02, wenn Ihr Astra Trident in einem Fenster mit vier Versionen von Version 24.02 angezeigt wird. Sie können beispielsweise direkt von Astra Trident 23.04 auf Astra Control Provisioner 24.02 aktualisieren.
   - Wenn Sie Astra Trident 23.10 oder höher verwenden, stellen Sie sicher, dass es für Astra Control Provisioner verwendet wurde "Aktiviert". Astra Control Provisioner kann nicht mit Versionen von Astra Control Center vor 23.10 verwendet werden. "Upgrade für die Astra Control Provisioner" Da es nun dieselbe Version wie das Astra Control Center hat, stellen Sie ein Upgrade auf die neuesten Funktionen bereit.
3. Stellen Sie sicher, dass alle Pods (einschließlich `trident-acp`) Läuft:

```
kubectl get pods -n trident
```

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

```
kubectl get sc
```

Beispielantwort:

```
NAME                   PROVISIONER            RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete         Immediate
true                   5d23h
```

Erstellen Sie eine Clusterrolle kubeconfig

Für Cluster, die mit kubeconfig gemanagt werden, können Sie optional eine Administratorrolle mit eingeschränkten Berechtigungen oder erweiterten Berechtigungen für Astra Control Center erstellen. Dies ist kein erforderliches Verfahren für das Astra Control Center-Setup, da Sie bereits einen kubeconfig als Teil des konfiguriert haben "Installationsprozess".

Dieses Verfahren hilft Ihnen, ein separates kubeconfig zu erstellen, wenn eines der folgenden Szenarien auf Ihre Umgebung zutrifft:

- Sie möchten die Astra Control-Berechtigungen auf die Cluster beschränken, die sie verwaltet
- Sie verwenden mehrere Kontexte und können nicht den Standard Astra Control kubeconfig verwenden, der während der Installation konfiguriert wurde, oder eine eingeschränkte Rolle mit einem einzelnen Kontext funktioniert nicht in Ihrer Umgebung

Bevor Sie beginnen

Stellen Sie sicher, dass Sie für den Cluster, den Sie verwalten möchten, vor dem Ausführen der Schritte des Verfahrens Folgendes haben:

- Kubectl v1.23 oder höher installiert
- Kubectl Zugriff auf den Cluster, den Sie mit Astra Control Center hinzufügen und verwalten möchten

> Bei diesem Verfahren benötigen Sie keinen kubectl-Zugriff auf den Cluster, auf dem Astra Control Center ausgeführt wird.

- Ein aktiver kubeconfig für den Cluster, den Sie mit Clusteradministratorrechten für den aktiven Kontext verwalten möchten

Schritte

1. Service-Konto erstellen:
   a. Erstellen Sie eine Dienstkontendatei mit dem Namen `astracontrol-service-account.yaml`.

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

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

b. Wenden Sie das Servicekonto an:

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

2. Erstellen Sie eine der folgenden Clusterrollen mit ausreichenden Berechtigungen für ein Cluster, das von Astra Control gemanagt werden kann:

Eingeschränkte Cluster-Rolle

Diese Rolle enthält die Mindestberechtigungen, die für das Management eines Clusters durch Astra Control erforderlich sind:

a. Erstellen Sie ein `ClusterRole` Datei mit dem Namen, z. B. `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
```

```
    - deployments
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
    - volumesnapshotcontents
    verbs:
    - delete

  # Watch resources
  # Necessary to monitor progress
  - apiGroups:
    - ""
    resources:
    - pods
    - replicationcontrollers
    - replicationcontrollers/scale
    verbs:
    - watch

  # Update resources
  - apiGroups:
    - ""
    - build.openshift.io
    - image.openshift.io
    resources:
    - builds/details
    - replicationcontrollers
    - replicationcontrollers/scale
```

```
  - imagestreams/layers
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Nur für OpenShift-Cluster) Anhängen Sie am Ende des an `astra-admin-account.yaml` Datei:

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. Wenden Sie die Cluster-Rolle an:

```
kubectl apply -f astra-admin-account.yaml
```

Erweiterte Cluster-Rolle

Diese Rolle enthält erweiterte Berechtigungen für ein Cluster, das von Astra Control gemanagt werden kann. Sie können diese Rolle verwenden, wenn Sie mehrere Kontexte verwenden und nicht den während der Installation konfigurierten Astra Control kubeconfig verwenden können oder eine eingeschränkte Rolle mit einem einzelnen Kontext in Ihrer Umgebung nicht funktioniert:

> Im Folgenden `ClusterRole` Schritte sind ein allgemeines Kubernetes-Beispiel. Anweisungen zu Ihrer spezifischen Umgebung finden Sie in der Dokumentation zur Kubernetes-Distribution.

a. Erstellen Sie ein `ClusterRole` Datei mit dem Namen, z. B. `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. Wenden Sie die Cluster-Rolle an:

```
kubectl apply -f astra-admin-account.yaml
```

3. Erstellen Sie die Cluster-Rolle, die für die Cluster-Rolle an das Service-Konto gebunden ist:

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

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

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

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

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

4. Erstellen und Anwenden des Token-Geheimnisses:
   a. Erstellen Sie eine Geheimdatei mit dem Namen Token `secret-astracontrol-service-account.yaml`.

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

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. Wenden Sie den Token-Schlüssel an:

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

5. Fügen Sie dem Dienstkonto den Token-Schlüssel hinzu, indem Sie den Namen dem hinzufügen `secrets` Array (die letzte Zeile im folgenden Beispiel):

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. 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-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Die Indizes für jedes Element im `secrets` Array beginnt mit 0. Im obigen Beispiel der Index für `astracontrol-service-account-dockercfg-48xhx` Wäre 0 und der Index für `secret-astracontrol-service-account` Sind es 1. Notieren Sie sich in Ihrer Ausgabe die Indexnummer für den Geheimschlüssel des Dienstkontos. Im nächsten Schritt benötigen Sie diese Indexnummer.

7. Erzeugen Sie den kubeconfig wie folgt:
   a. Erstellen Sie ein `create-kubeconfig.sh` Datei:
   b. Austausch `TOKEN_INDEX` Am Anfang des folgenden Skripts mit dem korrekten Wert.

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

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. 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'

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
```

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

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

8. (Optional) Umbenennen Sie die kubeconfig auf einen aussagekräftigen Namen für Ihr Cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

(Tech Preview) Installieren Sie Astra Connector für gemanagte Cluster

Von Astra Control Center gemanagte Cluster ermöglichen mithilfe von Astra Connector die Kommunikation zwischen dem gemanagten Cluster und Astra Control Center. Sie müssen Astra Connector auf allen Clustern installieren, die Sie verwalten möchten.

Astra Connector Installieren

Sie installieren Astra Connector mithilfe von Kubernetes-Befehlen und CR-Dateien (Custom Resource).

Über diese Aufgabe

- Wenn Sie diese Schritte ausführen, führen Sie diese Befehle auf dem Cluster aus, den Sie mit Astra Control managen möchten.
- Wenn Sie einen Bastion-Host verwenden, geben Sie diese Befehle über die Befehlszeile des Bastion-Hosts aus.

Bevor Sie beginnen

- Sie benötigen Zugriff auf den Cluster, den Sie mit Astra Control managen möchten.

- Sie benötigen Kubernetes-Administratorberechtigungen, um den Astra Connector Operator auf dem Cluster zu installieren.

> Wenn das Cluster mit der Durchsetzung der Pod-Sicherheitszulassung konfiguriert ist, was der Standard für Kubernetes-Cluster ab Version 1.25 ist, müssen Sie die PSA-Einschränkungen für die entsprechenden Namespaces aktivieren. Siehe "Bereiten Sie Ihre Umgebung mit Astra Control auf das Cluster-Management vor" Weitere Anweisungen.

Schritte

1. Installieren Sie den Astra Connector Operator auf dem Cluster, den Sie mit Astra Control managen möchten. Wenn Sie diesen Befehl ausführen, wird der Namespace verwendet `astra-connector-operator` Wird erstellt und die Konfiguration wird auf den Namespace angewendet:

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. Überprüfen Sie, ob der Bediener installiert und bereit ist:

```
kubectl get all -n astra-connector-operator
```

3. Holen Sie sich ein API-Token von Astra Control. Siehe "Dokumentation von Astra Automation" Weitere Anweisungen.
4. Erstellen Sie mithilfe des Tokens einen Schlüssel. Ersetzen Sie <API_TOKEN> durch das Token, das Sie von Astra Control erhalten haben:

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Erstellen Sie einen Docker-Schlüssel, um das Astra Connector-Image zu übertragen. Ersetzen Sie Werte in Klammern <> durch Informationen aus Ihrer Umgebung:

> Die <ASTRA_CONTROL_ACCOUNT_ID> finden Sie in der Web-UI von Astra Control. Wählen Sie in der Web-Benutzeroberfläche das Symbol oben rechts auf der Seite aus und wählen Sie **API Access**.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Erstellen Sie die Astra Connector CR-Datei und benennen Sie sie `astra-connector-cr.yaml`.

Aktualisieren Sie die Werte in Klammern <>, um sie an die Astra Control-Umgebung und die Cluster-Konfiguration anzupassen:

- <ASTRA_CONTROL_ACCOUNT_ID>: Erhalten von der Astra Control Web-UI während des vorhergehenden Schritts.
- <CLUSTER_NAME>: Der Name, dem dieser Cluster in Astra Control zugewiesen werden soll.
- <ASTRA_CONTROL_URL>: Die Web UI URL von Astra Control. Beispiel:

```
https://astra.control.url
```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. Nachdem Sie das ausgefüllt haben `astra-connector-cr.yaml` Datei mit den richtigen Werten, CR anwenden:

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Überprüfen Sie, ob der Astra Connector vollständig bereitgestellt ist:

```
kubectl get all -n astra-connector
```

9. Überprüfen Sie, ob das Cluster bei Astra Control registriert ist:

```
kubectl get astraconnectors.astra.netapp.io -A
```

Sie sollten eine Ausgabe wie die folgende sehen:

```
NAMESPACE         NAME              REGISTERED   ASTRACONNECTORID
STATUS
astra-connector   astra-connector   true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Überprüfen Sie, ob der Cluster in der Liste der verwalteten Cluster auf der Seite **Cluster** der Astra Control Web UI angezeigt wird.

Fügen Sie einen Cluster hinzu

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.

Bevor Sie beginnen

- Bevor Sie ein Cluster hinzufügen, überprüfen und führen Sie die erforderlichen Maßnahmen durch "Erforderliche Aufgaben".
- Wenn Sie einen ONTAP SAN-Treiber verwenden, stellen Sie sicher, dass Multipath auf allen Kubernetes-Clustern aktiviert ist.

Schritte

1. Navigieren Sie entweder über das Dashboard oder über das Menü Cluster:
   - Wählen Sie in der Ressourcenübersicht aus **Dashboard** im Bereich Cluster die Option **Hinzufügen** aus.
   - Wählen Sie im linken Navigationsbereich **Cluster** und dann auf der Seite Cluster **Cluster hinzufügen** 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: Wenn Sie ein kubeconfig für eine eingeschränkte Clusterrolle erstellt haben, die mit verwendet wird "Diesem Prozess dar", Vergewissern Sie sich, dass in diesem Schritt kubeconfig hochgeladen oder eingefügt wird.

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

> Wählen Sie eine Storage-Klasse aus, die in der Astra Control Provisioner-Konfiguration und mit ONTAP Storage konfiguriert ist.

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

Ergebnis

Der Cluster wechselt in den **Entdeckungs**-Zustand und dann in **gesund**. Sie managen jetzt das Cluster mit dem 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. Sie müssen in den Überwachungsprotokollen nachsehen, um das Problem zu beheben.

Aktivieren Sie die Authentifizierung auf einem ONTAP Storage Back-End

Astra Control Center bietet zwei Arten der Authentifizierung eines ONTAP-Backends:

- **Credential-basierte Authentifizierung**: Der Benutzername und das Passwort an einen ONTAP-Benutzer mit den erforderlichen Berechtigungen. Sie sollten eine vordefinierte Sicherheits-Login-Rolle wie admin oder vsadmin verwenden, um maximale Kompatibilität mit ONTAP-Versionen zu gewährleisten.
- **Zertifikatbasierte Authentifizierung**: Astra Control Center kann auch mit einem ONTAP-Cluster kommunizieren, indem ein auf dem Backend installiertes Zertifikat verwendet wird. Verwenden Sie gegebenenfalls das Clientzertifikat, den Schlüssel und das Zertifikat der vertrauenswürdigen Zertifizierungsstelle (empfohlen).

Sie können später vorhandene Back-Ends aktualisieren, um von einem Authentifizierungstyp zu einer anderen zu wechseln. Es wird jeweils nur eine Authentifizierungsmethode unterstützt.

Aktivieren Sie die Anmeldeinformationsbasierte Authentifizierung

Astra Control Center erfordert die Anmeldeinformationen für einen Cluster-Scoped `admin` Zur Kommunikation mit dem ONTAP-Backend. Sie sollten standardmäßige, vordefinierte Rollen wie verwenden `admin`. So wird die

Kompatibilität mit zukünftigen ONTAP Versionen sichergestellt, für die Funktionskompatibilität für zukünftige Astra Control Center Versionen zur Verfügung stehen könnte.

> Eine benutzerdefinierte Sicherheits-Login-Rolle kann erstellt und mit Astra Control Center verwendet werden, wird aber nicht empfohlen.

Eine Beispiel-Backend-Definition sieht so aus:

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

Die Backend-Definition ist der einzige Ort, an dem die Anmeldeinformationen im Klartext gespeichert werden. Die Erstellung oder Aktualisierung eines Backend ist der einzige Schritt, der Kenntnisse über die Anmeldeinformationen erfordert. Daher handelt es sich um einen reinen Admin-Vorgang, der vom Kubernetes- oder Storage-Administrator ausgeführt werden kann.

Aktivieren Sie die zertifikatbasierte Authentifizierung

Astra Control Center kann mithilfe von Zertifikaten mit neuen und vorhandenen ONTAP Back-Ends kommunizieren. Geben Sie die folgenden Informationen in die Backend-Definition ein.

- `clientCertificate`: Kundenzertifikat.
- `clientPrivateKey`: Zugehöriger privater Schlüssel.
- `trustedCACertificate`: Trusted CA-Zertifikat. Bei Verwendung einer vertrauenswürdigen CA muss dieser Parameter angegeben werden. Dies kann ignoriert werden, wenn keine vertrauenswürdige CA verwendet wird.

Sie können einen der folgenden Zertifikatstypen verwenden:

- Selbstsigniertes Zertifikat
- Drittanbieter-Zertifikat

Aktivieren Sie die Authentifizierung mit einem selbstsignierten Zertifikat

Ein typischer Workflow umfasst die folgenden Schritte.

Schritte

1. Erzeugen eines Clientzertifikats und eines Schlüssels. Legen Sie beim Generieren den allgemeinen Namen (Common Name, CN) auf den ONTAP-Benutzer fest, der sich als authentifizieren soll.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
```

2. Installieren Sie das Clientzertifikat des Typs `client-ca` Und drücken Sie auf dem ONTAP-Cluster.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Vergewissern Sie sich, dass die ONTAP-Sicherheits-Anmeldungsrolle die Zertifikatauthentifizierung unterstützt.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Testen Sie die Authentifizierung mithilfe des generierten Zertifikats. Ersetzen Sie <ONTAP Management LIF> und <vserver name> durch die Management-LIF-IP und den SVM-Namen. Sie müssen sicherstellen, dass die Service-Richtlinie für das LIF auf festgelegt ist `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. Fügen Sie mithilfe der Werte aus dem vorherigen Schritt das Speicher-Backend in der Astra Control Center-Benutzeroberfläche hinzu.

Aktivieren Sie die Authentifizierung mit einem Zertifikat eines Drittanbieters

Wenn Sie über ein Zertifikat eines Drittanbieters verfügen, können Sie mit diesen Schritten eine zertifikatbasierte Authentifizierung einrichten.

Schritte

1. Privaten Schlüssel und CSR generieren:

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>”
```

2. Leiten Sie die CSR an die Windows-Zertifizierungsstelle (Drittanbieter-CA) weiter, und stellen Sie das

signierte Zertifikat aus.

3. Laden Sie das signierte Zertifikat herunter und benennen Sie es mit `ontap_signed_cert.crt'.
4. Exportieren Sie das Stammzertifikat aus der Windows-CA (Drittanbieter-CA).
5. Benennen Sie diese Datei `ca_root.crt`

   Sie haben nun die folgenden drei Dateien:

   - **Privatschlüssel**: `ontap_signed_request.key` (Dies ist der entsprechende Schlüssel für das Serverzertifikat in ONTAP. Sie wird bei der Installation des Serverzertifikats benötigt.)
   - **Signiertes Zertifikat**: `ontap_signed_cert.crt` (Dies wird in ONTAP auch als *Server-Zertifikat* bezeichnet.)
   - **Stammzertifizierungsstelle**: `ca_root.crt` (In ONTAP wird dies auch als *Server-CA-Zertifikat* bezeichnet.)
6. Installieren Sie diese Zertifikate in ONTAP. Generieren und installieren `server` Und `server-ca` Zertifikate auf ONTAP.

Erweitern für Sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Erstellen Sie das Clientzertifikat für denselben Host für die passwortlose Kommunikation. Astra Control Center kommuniziert anhand dieses Verfahrens mit ONTAP.
8. Generieren und installieren Sie die Clientzertifikate auf ONTAP:

Erweitern für Sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate's CA and serial number for reference:

CA:
serial:
The certificate's generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```


9. Fügen Sie das Storage-Backend in der Astra Control Center-Benutzeroberfläche hinzu und geben Sie die folgenden Werte an:
   - **Client-Zertifikat**: ontap_Test_Client.pem
   - **Private Key**: ontap_test_client.key
   - **Vertrauenswürdiges CA-Zertifikat**: ontap_Signed_cert.crt

Fügen Sie ein Storage-Back-End hinzu

Nachdem Sie die Anmeldeinformationen oder Zertifikatauthentifizierungsinformationen eingerichtet haben, können Sie ein vorhandenes ONTAP-Storage-Back-End zu Astra Control Center hinzufügen, um seine Ressourcen zu managen.

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.

Das Hinzufügen und Managen von ONTAP Storage-Back-Ends in Astra Control Center ist optional, wenn Sie die NetApp SnapMirror Technologie verwenden, sofern Sie Astra Control Provisioner aktiviert haben.

Schritte

1. Wählen Sie im Dashboard im linken Navigationsbereich **Backend** aus.
2. Wählen Sie **Hinzufügen**.
3. Wählen Sie im Bereich vorhandene verwenden auf der Seite Speicher-Backend hinzufügen **ONTAP** aus.
4. Wählen Sie eine der folgenden Optionen:
   - **Administrator-Anmeldeinformationen verwenden**: Geben Sie die ONTAP Cluster Management IP-Adresse und die Admin-Anmeldeinformationen ein. Die Anmeldedaten müssen Cluster-weite Anmeldedaten aufweisen.

     > Der Benutzer, dessen Anmeldeinformationen Sie hier eingeben, muss über den verfügen `ontapi` Aktivieren der Zugriffsmethode für die Anmeldung beim Benutzer in ONTAP System Manager auf dem ONTAP Cluster. Wenn Sie Vorhaben, SnapMirror Replizierung zu verwenden, wenden Sie Benutzeranmeldeinformationen auf die Rolle „Admin“ an, die über die Zugriffsmethoden verfügt `ontapi` Und `http`, Auf Quell- und Ziel-ONTAP Clustern. Siehe "Managen von Benutzerkonten in der ONTAP Dokumentation" Finden Sie weitere Informationen.

   - **Ein Zertifikat** verwenden: Das Zertifikat hochladen `.pem` Datei, dem Zertifikatschlüssel `.key` Datei und optional die Zertifizierungsdatei.
5. Wählen Sie **Weiter**.
6. Bestätigen Sie die Backend-Details und wählen Sie **Verwalten**.

Ergebnis

Das Backend wird im angezeigt `online` Status in der Liste mit Zusammenfassungsinformationen.

> Möglicherweise müssen Sie die Seite aktualisieren, damit das Backend angezeigt wird.

Fügen Sie einen Bucket hinzu

Sie können einen Bucket über die Astra Control UI oder hinzufügen "Astra Control API". 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 Ihre Applikationskonfiguration und Ihren persistenten Storage im selben Cluster klonen, benötigen Sie in Astra Control keinen Bucket. Für die Funktionalität von Applikations-Snapshots ist kein Bucket erforderlich.

Bevor Sie beginnen

- Stellen Sie sicher, dass ein Bucket vorhanden ist, der von den von Astra Control Center gemanagten Clustern erreichbar ist.
- Stellen Sie sicher, dass Sie über Anmeldedaten für den Bucket verfügen.
- Stellen Sie sicher, dass es sich bei dem Bucket um einen der folgenden Typen handelt:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Microsoft Azure
- Allgemein S3

> Amazon Web Services (AWS) und Google Cloud Platform (GCP) verwenden den Bucket-Typ Generic S3.

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

Schritte

1. Wählen Sie im linken Navigationsbereich **Buckets** aus.
2. Wählen Sie **Hinzufügen**.
3. Wählen Sie den Bucket-Typ aus.

> Wenn Sie einen Bucket hinzufügen, wählen Sie den richtigen Bucket-Provider aus und geben die richtigen Anmeldedaten für diesen Provider an. Beispielsweise akzeptiert die UI NetApp ONTAP S3 als Typ und akzeptiert StorageGRID-Anmeldedaten. Dies führt jedoch dazu, dass alle künftigen Applikations-Backups und -Wiederherstellungen, die diesen Bucket verwenden, fehlschlagen.

4. Geben Sie einen vorhandenen Bucket-Namen und eine optionale Beschreibung ein.

> Der Name und die Beschreibung des Buckets werden als Backupspeicherort angezeigt, den Sie später bei der Erstellung eines Backups auswählen können. Der Name wird auch während der Konfiguration der Schutzrichtlinien angezeigt.

5. Geben Sie den Namen oder die IP-Adresse des S3-Endpunkts ein.
6. Wählen Sie unter **Anmeldeinformationen auswählen** die Registerkarte **Hinzufügen** oder **vorhandene verwenden**.
   - Wenn Sie sich für **Hinzufügen** entschieden haben:
     i. Geben Sie einen Namen für die Anmeldedaten ein, der sie von anderen Anmeldeinformationen in Astra Control unterscheidet.
     ii. Geben Sie die Zugriffs-ID und den geheimen Schlüssel ein, indem Sie den Inhalt aus der Zwischenablage einfügen.
   - Wenn Sie sich für **vorhandenes** verwenden:
     i. Wählen Sie die vorhandenen Anmeldedaten aus, die Sie mit dem Bucket verwenden möchten.
7. Wählen Sie `Add`.

> Wenn Sie einen Bucket hinzufügen, markiert Astra Control einen Bucket mit der Standard-Bucket-Anzeige. Der erste von Ihnen erstellte Bucket wird der Standard-Bucket. Wenn Sie Buckets hinzufügen, können Sie sich später entscheiden "Legen Sie einen weiteren Standard-Bucket fest".