Los geht's

Astra Control Center

NetApp
August 11, 2025

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

Inhalt

Los geht’s . . . 1
Weitere Informationen zu Astra Control . . . 1
Funktionen . . . 1
Implementierungsmodelle . . . 1
Funktionsweise des Astra Control Service . . . 3
So funktioniert Astra Control Center . . . 4
Finden Sie weitere Informationen . . . 5
Anforderungen des Astra Control Centers . . . 5
Unterstützte Host-Cluster-Kubernetes-Umgebungen . . . 5
Ressourcenanforderungen des Host-Clusters . . . 6
Service-Mesh-Anforderungen . . . 7
Astra Trident . . . 7
Astra Control Provisioner . . . 7
Storage-Back-Ends . . . 7
Astra Control Center-Lizenz . . . 8
Netzwerkanforderungen . . . 9
Ingress für lokale Kubernetes Cluster . . . 10
Unterstützte Webbrowser . . . 10
Zusätzliche Anforderungen an Applikations-Cluster . . . 10
Wie es weiter geht . . . 11
Schnellstart für Astra Control Center . . . 11
Finden Sie weitere Informationen . . . 12
Übersicht über die Installation . . . 12
Installieren Sie das Astra Control Center mithilfe des Standardprozesses . . . 12
Installieren Sie Astra Control Center mit OpenShift OperatorHub . . . 52
Installieren Sie Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend . . . 63
Konfigurieren Sie nach der Installation das Astra Control Center . . . 75
Einrichten des Astra Control Center . . . 81
Fügen Sie eine Lizenz für Astra Control Center hinzu . . . 81
Astra Control Provisioner Aktivieren . . . 82
Bereiten Sie Ihre Umgebung mit Astra Control auf das Cluster-Management vor . . . 92
(Tech Preview) Installieren Sie Astra Connector für gemanagte Cluster . . . 104
Fügen Sie einen Cluster hinzu . . . 107
Aktivieren Sie die Authentifizierung auf einem ONTAP Storage Back-End . . . 108
Fügen Sie ein Storage-Back-End hinzu . . . 115
Fügen Sie einen Bucket hinzu . . . 116

Los geht’s

Weitere Informationen zu Astra Control

Astra Control ist eine Kubernetes-Lösung für das Lifecycle-Management von Applikationsdaten, die den Betrieb zustandsorientierte Applikationen vereinfacht. Einfacher Schutz, Backup, Replizierung und Migration von Kubernetes-Workloads und sofortige Erstellung von Applikationsklonen

Funktionen

Astra Control bietet entscheidende Funktionen für das Lifecycle Management von Kubernetes-Applikationsdaten:

- Automatisches Management von persistentem Storage
- Erstellen Sie applikationsorientierte Snapshots und Backups nach Bedarf
- Automatisierung von richtlinienbasierten Snapshot- und Backup-Vorgängen
- Migrieren Sie Applikationen und Daten von einem Kubernetes-Cluster zu einem anderen
- Replizieren von Applikationen auf ein Remote-System mit NetApp SnapMirror Technologie (Astra Control Center)
- Klonen von Applikationen von Staging hin zur Produktion
- Darstellung des Anwendungszustands und des Schutzstatus
- Verwenden Sie eine Web-Oberfläche oder eine API zur Implementierung Ihrer Backup- und Migration-Workflows

Implementierungsmodelle

Astra Control ist in zwei Implementierungsmodellen erhältlich:

- **Astra Control Service**: Ein von NetApp gemanagter Service, der applikationskonsistentes Datenmanagement von Kubernetes Clustern in Umgebungen mehrerer Cloud-Provider sowie selbst gemanagte Kubernetes Cluster bietet.
- **Astra Control Center**: Gemanagte Software für applikationsgerechtes Datenmanagement von Kubernetes-Clustern, die in Ihrer On-Premises-Umgebung ausgeführt werden. Astra Control Center kann auch auf mehreren Cloud-Provider-Umgebungen mit einem NetApp Cloud Volumes ONTAP Storage-Backend installiert werden.

	Astra Control Service	Astra Control Center
Wie wird das angeboten?	Vollständig gemanagter Cloud-Service von NetApp	Als Software, die Sie herunterladen, installieren und verwalten können
Wo wird sie gehostet?	In einer Public Cloud von NetApp ihrer Wahl	In Ihrem eigenen Kubernetes-Cluster
Wie wird sie aktualisiert?	Gemanagt von NetApp	Sie verwalten jegliche Updates

	Astra Control Service	Astra Control Center
Welche Kubernetes-Distributionen werden unterstützt?	• **Cloud-Provider** ◦ Amazon Web Services ▪ Amazon Elastic Kubernetes Service (EKS) ◦ Google Cloud ▪ Google Kubernetes Engine (GKE) ◦ Microsoft Azure ▪ Azure Kubernetes-Service (AKS) • **Selbstverwaltete Cluster** ◦ Kubernetes (Vorgelagert) ◦ Rancher Kubernetes Engine (RKE) ◦ Red hat OpenShift Container Platform • **On-Premises-Cluster** ◦ Lokale Red hat OpenShift Container-Plattform	• Azure Kubernetes Service für Azure Stack HCI • Google Anthos • Kubernetes (Vorgelagert) • Rancher Kubernetes Engine (RKE) • Red hat OpenShift Container Platform

	Astra Control Service	Astra Control Center
Welche Storage-Back-Ends werden unterstützt?	• **Cloud-Provider** ◦ Amazon Web Services ▪ Amazon EBS ▪ Amazon FSX für NetApp ONTAP ▪ "Cloud Volumes ONTAP" ◦ Google Cloud ▪ Google Persistent Disk ▪ NetApp Cloud Volumes Service ▪ "Cloud Volumes ONTAP" ◦ Microsoft Azure ▪ Über Azure Gemanagte Festplatten ▪ Azure NetApp Dateien ▪ "Cloud Volumes ONTAP" • **Selbstverwaltete Cluster** ◦ Amazon EBS ◦ Über Azure Gemanagte Festplatten ◦ Google Persistent Disk ◦ "Cloud Volumes ONTAP" ◦ NetApp MetroCluster ◦ "Longhorn" • **On-Premises-Cluster** ◦ NetApp MetroCluster ◦ NetApp ONTAP AFF und FAS Systeme ◦ NetApp ONTAP Select ◦ "Cloud Volumes ONTAP" ◦ "Longhorn"	• NetApp ONTAP AFF und FAS Systeme • NetApp ONTAP Select • "Cloud Volumes ONTAP" • "Longhorn"

Funktionsweise des Astra Control Service

Astra Control Service ist ein von NetApp gemanagter Cloud-Service, der ständig verfügbar und mit den neuesten Funktionen aktualisiert ist. Verschiedene Komponenten unterstützen das Lifecycle-Management von Applikationsdaten.

Astra Control Service funktioniert auf hohem Niveau wie folgt:

- Starten Sie mit Astra Control Service, indem Sie Ihren Cloud-Provider einrichten und einen Astra Account anfordern.
  - Für GKE-Cluster verwendet der Astra Control Service "NetApp Cloud Volumes Service für Google Cloud" Oder Google Persistent Disks als Storage-Backend für Ihre persistenten Volumes.
  - Für AKS-Cluster nutzt der Astra Control Service "Azure NetApp Dateien" Oder von Azure gemanagte Festplatten als Storage-Backend für Ihre persistenten Volumes.
  - Für Amazon EKS-Cluster verwendet Astra Control Service "Amazon Elastic Block Store" Oder "Amazon FSX für NetApp ONTAP" Das Storage-Backend für Ihre persistenten Volumes
- Sie fügen Ihre ersten Kubernetes-Computing-Ressourcen in den Astra Control Service ein. Astra Control Service übernimmt dann Folgendes:
  - Erstellung eines Objektspeicher in Ihrem Cloud-Provider-Konto, an dem Backup-Kopien gespeichert werden

    In Azure erstellt Astra Control Service außerdem eine Ressourcengruppe, ein Storage-Konto und Schlüssel für den Blob-Container.
  - Erstellt eine neue Administratorrolle und ein Kubernetes-Servicekonto auf dem Cluster.
  - Installiert den Link../concepts/architecture#astra-control-components[astra Control Provisioner^] auf dem Cluster und erstellt eine oder mehrere Storage-Klassen.
  - Wenn Sie ein Storage-Angebot mit NetApp Cloud-Services als Storage-Back-End verwenden, stellt Astra Control Service persistente Volumes für Ihre Applikationen bereit. Wenn Sie von Amazon EBS oder Azure gemanagte Festplatten als Storage-Backend verwenden, müssen Sie einen Provider-spezifischen CSI-Treiber installieren. Installationsanweisungen finden Sie in "Einrichten von Amazon Web Services" Und "Richten Sie Microsoft Azure mit von Azure gemanagten Festplatten ein".
- An dieser Stelle können Sie Ihrem Cluster Apps hinzufügen. Persistente Volumes werden auf der neuen Standard-Storage-Klasse bereitgestellt.
- Anschließend verwalten Sie diese Applikationen mithilfe des Astra Control Service und erstellen Snapshots, Backups und Klone.

Mit dem kostenlosen Plan von Astra Control können Sie bis zu 10 Namespaces in Ihrem Konto verwalten. Wenn Sie mehr als 10 verwalten möchten, müssen Sie die Abrechnung durch ein Upgrade vom kostenlosen Plan auf den Premium-Plan einrichten.

So funktioniert Astra Control Center

Astra Control Center wird lokal in Ihrer eigenen Private Cloud ausgeführt.

Astra Control Center unterstützt Kubernetes-Cluster mit einer für die Astra Control Provisioner konfigurierten Storage-Klasse mit einem ONTAP Storage-Back-End.

In Astra Control Center sind begrenzte Monitoring- und Telemetriedaten (mit Kennzahlen von 7 Tagen) verfügbar und über offene metrische Endpunkte in native Kubernetes-Monitoring-Tools (z. B. Prometheus und Grafana) exportiert.

Astra Control Center ist vollständig in das Ecosystem von AutoSupport und Active IQ Digital Advisor (auch als digitaler Berater bekannt) integriert, um Benutzern und NetApp-Support Fehlerbehebungs- und Verwendungsinformationen zu bieten.

Sie können Astra Control Center mit einer eingebetteten 90-Tage-Evaluierungslizenz ausprobieren. Bei der Evaluierung von Astra Control Center können Sie Support über E-Mail- und Community-Optionen erhalten.

Zudem haben Sie über das Dashboard für den Produktsupport Zugriff auf Knowledgebase-Artikel und -Dokumentation.

Um Astra Control Center zu installieren und zu verwenden, müssen Sie sicher sein "Anforderungen".

Astra Control Center funktioniert auf hohem Niveau wie folgt:

- Sie installieren Astra Control Center in Ihrer lokalen Umgebung. Erfahren Sie mehr darüber, wie Sie "Installieren Sie Astra Control Center".
- Sie führen einige Setup-Aufgaben wie die folgenden aus:
  - Lizenzierung einrichten.
  - Fügen Sie den ersten Cluster hinzu.
  - Fügen Sie ein Storage-Back-End hinzu, das beim Hinzufügen des Clusters erkannt wird.
  - Fügen Sie einen Objektspeicher-Bucket hinzu, der Ihre Applikations-Backups speichert.

Erfahren Sie mehr darüber, wie Sie "Einrichten des Astra Control Center".

Sie können Applikationen zu Ihrem Cluster hinzufügen. Wenn auch einige Applikationen bereits im Cluster gemanagt werden, können Sie sie mit dem Astra Control Center managen. Nutzen Sie dann das Astra Control Center, um Snapshots, Backups, Klone und Replizierungsbeziehungen zu erstellen.

Finden Sie weitere Informationen

- "Dokumentation des Astra Control Service"
- "Astra Control Center-Dokumentation"
- "Astra Trident-Dokumentation"
- "Astra Control API-Dokumentation"
- "ONTAP-Dokumentation"

Anforderungen des Astra Control Centers

Prüfen Sie zunächst die Bereitschaft Ihrer Betriebsumgebung, Anwendungscluster, Applikationen, Lizenzen und Ihres Webbrowsers. Stellen Sie sicher, dass Ihre Umgebung für den Einsatz und Betrieb von Astra Control Center diese Anforderungen erfüllt.

Unterstützte Host-Cluster-Kubernetes-Umgebungen

Astra Control Center wurde mit den folgenden Kubernetes-Host-Umgebungen validiert:

> Stellen Sie sicher, dass die Kubernetes-Umgebung, für die Sie Astra Control Center hosten, die grundlegenden Ressourcenanforderungen erfüllt, die in der offiziellen Dokumentation der Umgebung aufgeführt sind.

Kubernetes-Distribution auf Host-Cluster	Unterstützte Versionen
Azure Kubernetes Service für Azure Stack HCI	Azure Stack HCI 21H2 und 22H2 mit AKS 1.24.11 bis 1.26.6

Kubernetes-Distribution auf Host-Cluster	Unterstützte Versionen
Google Anthos	1.15 bis 1.16 (siehe Anforderungen für Google Anthos Ingress)
Kubernetes (Vorgelagert)	1.27 bis 1.29
Rancher Kubernetes Engine (RKE)	RKE 1: Versionen 1.24.17, 1.25.13, 1.26.8 mit Rancher Manager 2.7.9 RKE 2: Versionen 1.23.16 und 1.24.13 mit Rancher Manager 2.6.13 RKE 2: Versionen 1.24.17, 1.25.14, 1.26.9 mit Rancher Manager 2.7.9
Red hat OpenShift Container Platform	4.12 bis 4.14

Ressourcenanforderungen des Host-Clusters

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

> 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.

- **CPU-Erweiterungen**: Die CPUs in allen Knoten der Hosting-Umgebung müssen AVX-Erweiterungen aktiviert haben.
- **Worker Nodes**: Insgesamt mindestens 3 Worker Nodes, mit 4 CPU Cores und je 12 GB RAM
- **Anforderungen für den VMware Tanzu Kubernetes Grid Cluster**: Beachten Sie bei der Hosting von Astra Control Center auf einem VMware Tanzu Kubernetes Grid (TKG)- oder Tanzu Kubernetes Grid Integrated Edition (TKGi)-Cluster die folgenden Überlegungen.
  - Das standardmäßige VMware TKG- und TKGi-Konfigurationstoken läuft zehn Stunden nach der Bereitstellung ab. Wenn Sie Tanzu Portfolio-Produkte verwenden, müssen Sie eine Tanzu Kubernetes Cluster-Konfigurationsdatei mit einem nicht auslaufenden Token generieren, um Verbindungsprobleme zwischen Astra Control Center und verwalteten Anwendungsclustern zu vermeiden. Anweisungen finden Sie unter "Die Produktdokumentation zu VMware NSX-T Data Center."
  - Verwenden Sie die `kubectl get nsxlbmonitors -A` Befehl, um zu sehen, ob bereits ein Service-Monitor für die Annahme von Ingress-Traffic konfiguriert ist. Wenn vorhanden, sollten Sie MetalLB nicht installieren, da der vorhandene Servicemonitor eine neue Load Balancer-Konfiguration außer Kraft setzt.
  - Deaktivieren Sie die Durchsetzung der Standardspeicherklasse TKG oder TKGi auf allen Anwendungsclustern, die von Astra Control verwaltet werden sollen. Sie können dies tun, indem Sie die bearbeiten `TanzuKubernetesCluster` Ressource auf dem Namespace-Cluster.
  - Beachten Sie bei der Implementierung von Astra Control Center in einer TKG oder TKGi Umgebung die spezifischen Anforderungen für die Astra Control Provisioner:
    - Das Cluster muss privilegierte Workloads unterstützen.
    - Der `--kubelet-dir` Flag sollte auf den Speicherort des kubelet-Verzeichnisses gesetzt werden. Standardmäßig ist dies `/var/vcap/data/kubelet`.
    - Festlegen der Kubelet-Position unter Verwendung `--kubelet-dir` Ist für Trident Operator, Helm

und bekannt `tridentctl` Implementierungen.

Service-Mesh-Anforderungen

Es wird dringend empfohlen, eine unterstützte Vanilla-Version des Istio Service Mesh auf dem Astra Control Center Host Cluster zu installieren. Siehe "Unterstützte Versionen" Für unterstützte Versionen von Istio. Markenversionen von Istio Service-Mesh, wie OpenShift Service Mesh, werden nicht mit Astra Control Center validiert.

Um Astra Control Center mit dem auf dem Host-Cluster installierten Istio-Service-Mesh zu integrieren, müssen Sie die Integration als Teil eines Astra Control Center vornehmen "Installation" Und nicht unabhängig von diesem Prozess.

Die Installation und Verwendung von Astra Control Center ohne Konfiguration eines Service-Mesh auf dem Host-Cluster kann schwerwiegende Auswirkungen auf die Sicherheit haben.

Astra Trident

Wenn Sie mit diesem Release Astra Trident anstelle von Astra Control Provisioner verwenden möchten, werden Astra Trident 23.04 und neuere Versionen unterstützt. Für Astra Control Center ist erforderlich Astra Control Provisioner In zukünftigen Versionen verfügbar.

Astra Control Provisioner

Um die erweiterten Storage-Funktionen von Astra Control Provisioner zu verwenden, müssen Sie Astra Trident 23.10 oder höher installieren und aktivieren "Funktionen für die Astra Control Provisioner". Zur Nutzung der neuesten Funktionen für die Astra Control-Bereitstellung benötigen Sie die aktuellsten Versionen von Astra Trident und Astra Control Center.

- **Mindestversion für Astra Control Provisioner zur Verwendung mit Astra Control Center**: Astra Control Provisioner 23.10 oder höher installiert und konfiguriert.

ONTAP-Konfiguration mit Astra Trident

- **Storage class**: Konfigurieren Sie mindestens eine Storage-Klasse auf dem Cluster. Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass es die einzige Storage-Klasse mit der Standardbezeichnung ist.
- **Speichertreiber und Workerknoten**: Stellen Sie sicher, dass Sie die Workerknoten in Ihrem Cluster mit den entsprechenden Speichertreibern konfigurieren, damit die Pods mit dem Backend-Speicher interagieren können. Astra Control Center unterstützt die folgenden ONTAP-Treiber von Astra Trident:
  - `ontap-nas`
  - `ontap-san`
  - `ontap-san-economy` (Applikationsreplizierung ist bei diesem Storage-Klassen-Typ nicht verfügbar)
  - `ontap-nas-economy` (Snapshots und Richtlinien zur Applikationsreplizierung sind bei diesem Storage-Klassen-Typ nicht verfügbar)

Storage-Back-Ends

Stellen Sie sicher, dass Sie ein unterstütztes Backend mit ausreichender Kapazität haben.

- **Erforderliche Back-End-Speicherkapazität**: Mindestens 500 GB verfügbar
- **Unterstützte Backends**: Astra Control Center unterstützt folgende Speicher-Backends:
  - NetApp ONTAP 9.9.1 oder höher AFF, FAS und ASA Systeme
  - NetApp ONTAP Select 9.9.1 oder höher
  - NetApp Cloud Volumes ONTAP 9.9.1 oder höher
  - (Zur technischen Vorschau für Astra Control Center) NetApp ONTAP 9.10.1 oder höher für Datensicherungsvorgänge als technische Vorschau
  - Longhorn 1.5.0 oder höher
    - Erfordert die manuelle Erstellung eines VolumeSnapshotClass-Objekts. Siehe "Longhorn-Dokumentation" Weitere Anweisungen.
  - NetApp MetroCluster
    - Verwaltete Kubernetes-Cluster müssen in einer Stretch-Konfiguration vorliegen.
  - Storage-Back-Ends bei unterstützten Cloud-Providern verfügbar

ONTAP-Lizenzen

Um Astra Control Center zu nutzen, müssen Sie je nach den Anforderungen die folgenden ONTAP-Lizenzen besitzen:

- FlexClone
- SnapMirror: Optional Nur für die Replizierung auf Remote-Systeme mit SnapMirror Technologie erforderlich. Siehe "Informationen zu SnapMirror Lizenzen".
- S3-Lizenz: Optional Nur für ONTAP S3 Buckets erforderlich

Informationen darüber, ob auf Ihrem ONTAP System die erforderlichen Lizenzen vorhanden sind, finden Sie unter "Managen Sie ONTAP Lizenzen".

NetApp MetroCluster

Wenn Sie NetApp MetroCluster als Storage-Backend verwenden, müssen Sie Folgendes tun:

- Geben Sie eine SVM-Management-LIF als Back-End-Option im von Ihnen verwendeten Astra Trident-Treiber an
- Stellen Sie sicher, dass Sie über die entsprechende ONTAP-Lizenz verfügen

Informationen zum Konfigurieren der logischen MetroCluster-Schnittstelle finden Sie in den folgenden Optionen und Beispielen für die einzelnen Treiber:

- "San"
- "NAS"

Astra Control Center-Lizenz

Für Astra Control Center ist eine Astra Control Center Lizenz erforderlich. Bei der Installation von Astra Control Center ist bereits eine eingebettete 90-Tage-Evaluierungslizenz für 4,800 CPU-Einheiten aktiviert. Wenn Sie mehr Kapazität oder andere Evaluierungsbedingungen benötigen, oder ein Upgrade auf eine komplette Lizenz wünschen, können Sie eine andere Evaluierungslizenz oder volle Lizenz von NetApp erhalten. Sie benötigen eine Lizenz zum Schutz Ihrer Applikationen und Daten.

Astra Control Center können Sie ausprobieren, indem Sie sich für eine kostenlose Testversion anmelden. Registrieren Sie sich "Hier".

Informationen zum Einrichten der Lizenz finden Sie unter "Verwenden Sie eine 90-Tage-Evaluierungslizenz".

Weitere Informationen zur Funktionsweise von Lizenzen finden Sie unter "Lizenzierung".

Netzwerkanforderungen

Konfigurieren Sie Ihre Betriebsumgebung so, dass Astra Control Center ordnungsgemäß kommunizieren kann. Die folgenden Netzwerkkonfigurationen sind erforderlich:

- **FQDN-Adresse**: Sie müssen eine FQDN-Adresse für Astra Control Center haben.
- **Zugang zum Internet**: Sie sollten festlegen, ob Sie Zugang zum Internet von außen haben. Wenn Sie dies nicht tun, sind einige Funktionen möglicherweise eingeschränkt, z. B. das Senden von Support-Paketen an das "NetApp Support Website".
- **Port Access**: Die Betriebsumgebung, die das Astra Control Center hostet, kommuniziert über die folgenden TCP-Ports. Sie sollten sicherstellen, dass diese Ports über beliebige Firewalls zugelassen sind, und Firewalls so konfigurieren, dass jeder HTTPS-ausgehenden Datenverkehr aus dem Astra-Netzwerk zugelassen wird. Einige Ports erfordern Verbindungen zwischen der Umgebung, in der Astra Control Center gehostet wird, und jedem verwalteten Cluster (sofern zutreffend).

Sie können Astra Control Center in einem Dual-Stack-Kubernetes-Cluster implementieren. Astra Control Center kann Applikationen und Storage-Back-Ends managen, die für den Dual-Stack-Betrieb konfiguriert wurden. Weitere Informationen zu Dual-Stack-Cluster-Anforderungen finden Sie im "Kubernetes-Dokumentation".

Quelle	Ziel	Port	Protokoll	Zweck
Client-PC	Astra Control Center	443	HTTPS	UI-/API-Zugriff – Stellen Sie sicher, dass dieser Port in beide Richtungen zwischen Astra Control Center und dem System offen ist, mit dem auf Astra Control Center zugegriffen wird
Kennzahlenverbraucher	Astra Control Center Worker-Node	9090	HTTPS	Kennzahlen Datenkommunikation - sicherstellen, dass jeder verwaltete Cluster auf diesen Port auf dem Cluster zugreifen kann, das Astra Control Center hostet (Kommunikation in zwei Bereichen erforderlich)

Quelle	Ziel	Port	Protokoll	Zweck
Astra Control Center	Amazon S3 Storage-Bucket-Provider	443	HTTPS	Amazon S3 Storage-Kommunikation
Astra Control Center	NetApp AutoSupport	443	HTTPS	Kommunikation zwischen NetApp AutoSupport
Astra Control Center	Gemanagter Kubernetes-Cluster	443/6443 **HINWEIS**: Der Port, den der verwaltete Cluster verwendet, kann je nach Cluster variieren. Informationen finden Sie in der Dokumentation des Anbieters der Cluster-Software.	HTTPS	Kommunikation mit dem verwalteten Cluster – Stellen Sie sicher, dass dieser Port auf beiden Wegen zwischen dem Cluster, der Astra Control Center hostet, und jedem verwalteten Cluster offen ist

Ingress für lokale Kubernetes Cluster

Sie können die Art der Netzwerk Ingress Astra Control Center verwendet wählen. Astra Control Center nutzt standardmäßig das Astra Control Center Gateway (Service/Trafik) als Cluster-weite Ressource. Astra Control Center unterstützt auch den Einsatz eines Service Load Balancer, sofern diese in Ihrer Umgebung zugelassen sind. Wenn Sie lieber einen Service-Load-Balancer verwenden und noch nicht eine konfiguriert haben, können Sie den MetalLB-Load-Balancer verwenden, um dem Dienst automatisch 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.

Der Load Balancer sollte eine IP-Adresse verwenden, die sich im gleichen Subnetz wie die IP-Adressen des Astra Control Center Worker-Knotens befindet.

Weitere Informationen finden Sie unter "Eindringen für den Lastenausgleich einrichten".

Anforderungen für Google Anthos Ingress

Beachten Sie beim Hosten von Astra Control Center auf einem Google Anthos Cluster, dass Google Anthos standardmäßig den MetalLB Load Balancer und den Istio Ingress Service enthält, sodass Sie während der Installation einfach die generischen Ingress-Funktionen von Astra Control Center verwenden können. Siehe "Installationsdokumentation für Astra Control Center" Entsprechende Details.

Unterstützte Webbrowser

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

Zusätzliche Anforderungen an Applikations-Cluster

Beachten Sie diese Anforderungen, wenn Sie die folgenden Funktionen des Astra Control Center nutzen möchten:

- **Anforderungen an den Anwendungscluster**: "Anforderungen für das Cluster-Management"

- **Verwaltete Anwendungsanforderungen**: "Anforderungen für das Applikationsmanagement"
- **Zusätzliche Anforderungen an die Anwendungsreplikation**: "Replikationsvoraussetzungen"

Wie es weiter geht

Sehen Sie sich die an "Schnellstart" Überblick.

Schnellstart für Astra Control Center

Hier finden Sie eine Übersicht über die Schritte, die für den Einstieg in das Astra Control Center erforderlich sind. Die Links in den einzelnen Schritten führen zu einer Seite, die weitere Details enthält.

Kubernetes-Cluster-Anforderungen prüfen

Stellen Sie sicher, dass Ihre Umgebung die folgenden Anforderungen erfüllt:

- Kubernetes Cluster*
- "Stellen Sie sicher, dass Ihr Host-Cluster die Anforderungen der Betriebsumgebung erfüllt"
- "Konfigurieren Sie Ingress für den Lastausgleich von lokalen Kubernetes-Clustern"

Storage-Integration

- "Stellen Sie sicher, dass Ihre Umgebung auch Astra Control Provisioner umfasst"
- "Aktivieren Sie Astra Control Provisioner für erweiterte Management- und Storage-Bereitstellungsfunktionen"
- "Cluster-Worker-Nodes vorbereiten"
- "Konfiguration von Storage-Back-Ends"
- "Konfigurieren von Speicherklassen"
- "Installieren Sie einen Volume-Snapshot-Controller"
- "Erstellen Sie eine Volume Snapshot-Klasse"

ONTAP-Anmeldedaten

- "Konfigurieren Sie die ONTAP-Anmeldedaten"

Laden Sie Astra Control Center herunter und installieren Sie es

Führen Sie die folgenden Installationsaufgaben aus:

- "Laden Sie Astra Control Center von der Download-Seite der NetApp Support-Website herunter"
- Beziehen Sie die NetApp Lizenzdatei:
  - Wenn Sie Astra Control Center evaluieren, ist bereits eine eingebettete Evaluierungslizenz enthalten
  - "Wenn Sie Astra Control Center bereits gekauft haben, generieren Sie Ihre Lizenzdatei"
- "Installieren Sie Astra Control Center"

- "Führen Sie weitere optionale Konfigurationsschritte durch"

3 Führen Sie einige erste Setup-Aufgaben aus

Führen Sie einige grundlegende Aufgaben aus, um zu beginnen:

- "Fügen Sie eine Lizenz hinzu"
- "Vorbereitung der Umgebung auf das Cluster Management"
- "Fügen Sie einen Cluster hinzu"
- "Fügen Sie ein Storage-Back-End hinzu"
- "Fügen Sie einen Bucket hinzu"

4 Nutzen Sie Das Astra Control Center

Nachdem Sie das Astra Control Center eingerichtet haben, verwenden Sie die Astra Control UI oder die "Astra Control API" So beginnen Sie mit der Verwaltung und dem Schutz von Apps:

- "Konten verwalten": Benutzer, Rollen, LDAP, Anmeldeinformationen und mehr.
- "Benachrichtigungen verwalten"
- "Applikationsmanagement": Ressourcen definieren, die verwaltet werden sollen.
- "Schützen von Applikationen": Schutzrichtlinien konfigurieren und Anwendungen replizieren, klonen und migrieren.

Finden Sie weitere Informationen

- "Verwenden Sie die Astra Control API"
- "Upgrade Astra Control Center"
- "Holen Sie sich Hilfe mit Astra Control"

Übersicht über die Installation

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

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

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

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

Installieren Sie das Astra Control Center mithilfe des Standardprozesses

Um Astra Control Center zu installieren, laden Sie die Installationsabbilder herunter und

führen Sie die folgenden Schritte aus. Mit diesem Verfahren können Sie Astra Control Center in Internet-angeschlossenen oder luftgekapderten Umgebungen installieren.

Eine Demonstration des Installationsvorgangs für Astra Control Center finden Sie unter "Dieses Video".

Bevor Sie beginnen

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

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

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

```
kubectl get apiservices
```

- **Stellen Sie einen routingfähigen FQDN sicher**: Der Astra FQDN, den Sie verwenden möchten, kann zum Cluster weitergeleitet werden. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.
- **Configure cert Manager**: Wenn ein cert Manager bereits im Cluster existiert, müssen Sie einige durchführen "Erforderliche Schritte" Damit Astra Control Center nicht versucht, seinen eigenen Cert Manager zu installieren. Standardmäßig installiert Astra Control Center während der Installation einen eigenen Cert-Manager.
- **(nur ONTAP-SAN-Treiber) Aktivieren Sie Multipath**: Wenn Sie einen ONTAP-SAN-Treiber verwenden, stellen Sie sicher, dass Multipath auf allen Kubernetes-Clustern aktiviert ist.

Sie sollten auch Folgendes berücksichtigen:

- **Zugriff auf die NetApp Astra Control Image Registry**:

  Sie haben die Möglichkeit, Installations-Images und Funktionserweiterungen für Astra Control, wie z. B. Astra Control Provisioner, aus der NetApp-Image-Registrierung zu beziehen.

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

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

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

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

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

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

Die Integration von Astra Control Center mit einem Service-Mesh ist nur während des Astra Control Center möglich "Installation" Und nicht unabhängig von diesem Prozess. Ein Wechsel von einer vernetzten in eine nicht vernetzte Umgebung wird nicht unterstützt.

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

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

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Schritte

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

- Laden Sie das Astra Control Center herunter und extrahieren Sie es
- wenn Sie eine lokale Registrierung verwenden
- Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen
- Installieren Sie den Operator Astra Control Center
- Konfigurieren Sie Astra Control Center
- Komplette Astra Control Center und Bedienerinstallation

- Überprüfen Sie den Systemstatus
- Eindringen für den Lastenausgleich einrichten
- Melden Sie sich in der UI des Astra Control Center an

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

Laden Sie das Astra Control Center herunter und extrahieren Sie es

Laden Sie die Bilder des Astra Control Center von einem der folgenden Standorte herunter:

- **Astra Control Service Image Registry**: Verwenden Sie diese Option, wenn Sie keine lokale Registrierung mit den Astra Control Center Images verwenden oder wenn Sie diese Methode dem Bundle-Download von der NetApp Support-Website vorziehen.
- **NetApp Support-Seite**: Verwenden Sie diese Option, wenn Sie eine lokale Registrierung mit den Astra Control Center-Images verwenden.

Astra Control-Image-Registrierung

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

NetApp Support Website

1. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`) Vom "Download-Seite für Astra Control Center".
2. (Empfohlen, aber optional) Laden Sie das Zertifikaten- und Unterschriftenpaket für Astra Control Center herunter (`astra-control-center-certs-[version].tar.gz`) Um die Signatur des Bündels zu überprüfen.

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

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

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

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

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

Führen Sie zusätzliche Schritte durch, wenn Sie eine lokale Registrierung verwenden

Wenn Sie planen, das Astra Control Center Bundle in Ihre lokale Registry zu schieben, müssen Sie das NetApp Astra kubectl Kommandozeilen-Plugin verwenden.

Installieren Sie das NetApp Astra kubectl Plug-in

Führen Sie diese Schritte aus, um das neueste NetApp Astra kubectl Kommandozeilen-Plugin zu installieren.

Bevor Sie beginnen

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

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

Schritte

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

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

   ```
   ls kubectl-astra/
   ```

2. Verschieben Sie die für Ihr Betriebssystem und die CPU-Architektur benötigte Datei in den aktuellen Pfad und benennen Sie sie in um `kubectl-astra`:

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

Fügen Sie die Bilder zu Ihrer Registrierung hinzu

1. Wenn Sie planen, das Astra Control Center-Paket in Ihre lokale Registrierung zu übertragen, führen Sie die entsprechende Schrittfolge für Ihre Container-Engine aus:

Docker

a. Wechseln Sie in das Stammverzeichnis des Tarballs. Sie sollten den sehen `acc.manifest.bundle.yaml` Datei und diese Verzeichnisse:

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

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

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

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

Podman

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

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

b. Melden Sie sich bei Ihrer Registrierung an:

```
podman login <YOUR_REGISTRY>
```

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

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

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

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Telefonbuch ändern:

```
cd manifests
```

Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen

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

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

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

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

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

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

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

Platzhalter `your_registry_path` Sollte die Position der Bilder, die Sie früher hochgeladen haben, entsprechen (z. B. `[Registry_URL]/netapp/astra/astracc/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-operator --docker-server=cr.astra.netapp.io --docker -username=[astra_account_id] --docker-password=[astra_api_token]
```

+

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

+

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

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

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

   b. Erstellen Sie ein Geheimnis für das `netapp-acc` (Oder Name des benutzerdefinierten Namespace).

Fügen Sie Docker-Informationen hinzu, und führen Sie je nach Registrierungseinstellung einen der entsprechenden Befehle aus:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=cr.astra.netapp.io --docker-username=[astra_account_id] --docker-password=[astra_api_token]
```

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

Installieren Sie den Operator Astra Control Center

1. (Nur lokale Registrierungsstellen) Wenn Sie eine lokale Registrierung verwenden, führen Sie die folgenden Schritte aus:

   a. Astra Control Center Operator Deployment YAML öffnen:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

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

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Ändern `ASTRA_IMAGE_REGISTRY` Für das `kube-rbac-proxy` Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben Vorheriger Schritt.

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
```

```
    control-plane: controller-manager
strategy:
  type: Recreate
template:
  metadata:
    labels:
      control-plane: controller-manager
  spec:
    containers:
    - args:
      - --secure-listen-address=0.0.0.0:8443
      - --upstream=http://127.0.0.1:8080/
      - --logtostderr=true
      - --v=10
      image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
      name: kube-rbac-proxy
      ports:
      - containerPort: 8443
        name: https
    - args:
      - --health-probe-bind-address=:8081
      - --metrics-bind-address=127.0.0.1:8080
      - --leader-elect
      env:
      - name: ACCOP_LOG_LEVEL
        value: "2"
      - name: ACCOP_HELM_INSTALLTIMEOUT
        value: 5m
      image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
      imagePullPolicy: IfNotPresent
      livenessProbe:
        httpGet:
          path: /healthz
          port: 8081
        initialDelaySeconds: 15
        periodSeconds: 20
      name: manager
      readinessProbe:
        httpGet:
          path: /readyz
          port: 8081
        initialDelaySeconds: 5
        periodSeconds: 10
      resources:
        limits:
          cpu: 300m
```

```
            memory: 750Mi
          requests:
            cpu: 100m
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

2. Installieren Sie den Astra Control Center-Operator:

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

Erweitern für Probenantwort:

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

3. Überprüfen Sie, ob Pods ausgeführt werden:

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

Konfigurieren Sie Astra Control Center

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

```
vim astra_control_center.yaml
```

Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

2. Ändern oder bestätigen Sie die folgenden Einstellungen:

AccountName

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

AstraVersion

Einstellung	Anleitung	Typ	Beispiel
`astraVersion`	Die zu implementierende Version des Astra Control Center: Für diese Einstellung ist keine Aktion erforderlich, da der Wert bereits ausgefüllt wird.	Zeichenfolge	`24.02.0-69`

AstraAddress

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

AutoSupport

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

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

E-Mail

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

Vorname

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

Nachname

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

ImageRegistry

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

Einstellung	Nutzung	Anleitung	Typ	Beispiel
`imageRegistry.name`	Erforderlich	Der Name der Astra Control-Image-Registrierung, die alle Images hostet, die für die Bereitstellung von Astra Control Center erforderlich sind. Der Wert wird vorausgefüllt, und es ist keine Aktion erforderlich, es sei denn, Sie haben eine lokale Registrierung konfiguriert. Ersetzen Sie bei einer lokalen Registrierung diesen vorhandenen Wert durch den Namen der Bildregistrierung, in die Sie die Bilder in der verschoben haben Vorheriger Schritt. Verwenden Sie es nicht `http://` Oder `https://` Im Registrierungsnamen.	Zeichenfolge	`cr.astra.netapp.io` (Standard) `example.registry.com/astra` (Beispiel für eine lokale Registrierung)

Einstellung	Nutzung	Anleitung	Typ	Beispiel
`imageRegistry.secret`	Optional	Der Name des Kubernetes Secret, das zur Authentifizierung mit der Bildregistrierung verwendet wird. Der Wert wird vorausgefüllt, und es ist keine Aktion erforderlich, es sei denn, Sie haben eine lokale Registrierung und den String konfiguriert, den Sie für diese Registrierung in eingegeben haben `imageRegistry.name` Erfordert ein Geheimnis. WICHTIG: Wenn Sie eine lokale Registrierung verwenden, die keine Autorisierung erfordert, müssen Sie diese löschen `secret` Zeile in `imageRegistry` Oder die Installation schlägt fehl.	Zeichenfolge	`astra-registry-cred`

Storage Class

Einstellung	Anleitung	Typ	Beispiel
`storageClass`	Ändern Sie das `storageClass` Wert von `ontap-gold` Auf eine andere Ressource von storageClass, wie von Ihrer Installation gefordert. Führen Sie den Befehl aus `kubectl get sc` So ermitteln Sie Ihre vorhandenen konfigurierten Speicherklassen. In die Manifestdatei muss eine für die Astra Control Provisioner konfigurierte Storage-Klasse eingegeben werden (`astra-control-center-<version>.manifest`) Und wird für Astra PVS verwendet. Wenn er nicht festgelegt ist, wird die Standard-Speicherklasse verwendet. HINWEIS: Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass diese die einzige Storage-Klasse mit der Standardbeschriftung ist.	Zeichenfolge	`ontap-gold`

VolumeReclaimPolicy

Einstellung	Anleitung	Typ	Optionen
`volumeReclaimPolicy`	Damit wird die Rückgewinnungsrichtlinie für die PVS von Astra festgelegt. Festlegen dieser Richtlinie auf `Retain` Behält persistente Volumes nach dem Löschen von Astra bei. Festlegen dieser Richtlinie auf `Delete` Löscht persistente Volumes nach dem Löschen von astra. Wenn dieser Wert nicht festgelegt ist, werden die PVS beibehalten.	Zeichenfolge	• `Retain` (Dies ist der Standardwert) • `Delete`

Typ

Einstellung	Anleitung	Typ	Optionen
`ingressType`	Verwenden Sie einen der folgenden Eingangstypen: **Allgemein** (`ingressType: "Generic"`) (Standard) Verwenden Sie diese Option, wenn Sie einen anderen Ingress-Controller verwenden oder Ihren eigenen Ingress-Controller verwenden möchten. Nachdem Astra Control Center bereitgestellt ist, müssen Sie das konfigurieren "Eingangs-Controller" Um Astra Control Center mit einer URL zu zeigen. WICHTIG: Wenn Sie ein Service-Mesh mit Astra Control Center verwenden möchten, müssen Sie auswählen `Generic` Als Eindringen Typ und richten Sie Ihre eigenen "Eingangs-Controller". **AccTraefik** (`ingressType: "AccTraefik"`) Verwenden Sie diese Option, wenn Sie keinen Ingress-Controller konfigurieren möchten. Dies implementiert das Astra Control Center `traefik` Gateway als Service des Typs Kubernetes Load Balancer: Astra Control Center nutzt einen Service vom Typ „loadbalancer“ (`svc/traefik` Im Astra Control Center	Zeichenfolge	• `Generic` (Dies ist der Standardwert) • `AccTraefik`

Skalengröße

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

AstraResourcesScaler

Einstellung	Anleitung	Typ	Optionen
`astraResourcesScaler`	Skalierungsoptionen für die Ressourcengrenzen von AstraControlCenter. Astra Control Center implementiert standardmäßig mit Ressourcenanfragen, die für die meisten Komponenten in Astra bereitgestellt werden. Mit dieser Konfiguration verbessert sich die Leistung des Astra Control Center Software-Stacks auch bei erhöhter Applikationslast und -Skalierung. In Szenarien mit kleineren Entwicklungs- oder Testclustern jedoch das CR-Feld `astraResourcesScalar` Kann auf festgelegt werden `Off`. Dadurch werden Ressourcenanforderungen deaktiviert und die Bereitstellung auf kleineren Clustern ist möglich.	Zeichenfolge	• `Default` (Dies ist der Standardwert) • `Off`

Zusätzliche Werte

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

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

crds

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

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

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

Beispiel für astra_Control_Center.yaml

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[cr.astra.netapp.io or your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

Komplette Astra Control Center und Bedienerinstallation

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

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

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

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

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

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

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

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

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

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

Überprüfen Sie den Systemstatus

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

Schritte

1. Vergewissern Sie sich, dass beim Installationsprozess keine Warnmeldungen zu den Validierungsprüfungen ausgegeben wurden:

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

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

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

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

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

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

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

Erweitern, um die Probenantwort zu erhalten

```
acc-helm-repo-5bd77c9ddd-8wxm2              1/1     Running     0
1h
activity-5bb474dc67-8l9ss                   1/1     Running     0
1h
activity-5bb474dc67-qbrtq                   1/1     Running     0
1h
api-token-authentication-6wbj2              1/1     Running     0
1h
api-token-authentication-9pgw6              1/1     Running     0
1h
api-token-authentication-tqf6d              1/1     Running     0
1h
asup-5495f44dbd-z4kft                       1/1     Running     0
1h
authentication-6fdd899858-5x45s             1/1     Running     0
1h
bucketservice-84d47487d-n9xgp               1/1     Running     0
1h
bucketservice-84d47487d-t5jhm               1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc               1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf               1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf    1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck    1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x        1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn        1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj                1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx                1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw    0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv            1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km            1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92       1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f          1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2          1/1     Running     0
1h
composite-volume-bff6f4f76-789nj            1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4            1/1     Running     0
1h
credentials-79fd64f788-m7m8f                1/1     Running     0
1h
credentials-79fd64f788-qnc6c                1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                 1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                 1/1     Running     0
1h
features-7b9585444c-7xd7m                   1/1     Running     0
1h
features-7b9585444c-dcqwc                   1/1     Running     0
1h
fluent-bit-ds-crq8m                         1/1     Running     0
1h
fluent-bit-ds-gmgq8                         1/1     Running     0
1h
fluent-bit-ds-gzr4f                         1/1     Running     0
1h
fluent-bit-ds-j6sf6                         1/1     Running     0
1h
fluent-bit-ds-v4t9f                         1/1     Running     0
1h
fluent-bit-ds-x7j59                         1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr             1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd             1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                 1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                 1/1     Running     0
1h
identity-95df98cb5-dvlmz                    1/1     Running     0
1h
identity-95df98cb5-krf59                    1/1     Running     0
1h
influxdb2-0                                 1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b          1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                    1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                    1/1     Running     0
1h
license-689cdd4595-2gsc8                    1/1     Running     0
1h
license-689cdd4595-g6vwk                    1/1     Running     0
1h
login-ui-57bb599956-4fwgz                   1/1     Running     0
1h
login-ui-57bb599956-rhztb                   1/1     Running     0
1h
loki-0                                      1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm             1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl             1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl        2/2     Running     0
1h
nats-0                                      1/1     Running     0
1h
nats-1                                      1/1     Running     0
1h
nats-2                                      1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd            1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                   1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                   1/1     Running     0
1h
openapi-864584bccc-75nlv                    1/1     Running     0
1h
openapi-864584bccc-zh6bx                    1/1     Running     0
1h
polaris-consul-consul-server-0              1/1     Running     0
1h
polaris-consul-consul-server-1              1/1     Running     0
1h
polaris-consul-consul-server-2              1/1     Running     0
1h
polaris-keycloak-0                          1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-815zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

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

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

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

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

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

Antwort:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  24.02.0-69
10.111.111.111  True
```

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

Eindringen für den Lastenausgleich einrichten

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

Nachdem Astra Control Center implementiert ist, müssen Sie den Ingress-Controller so konfigurieren, dass für das Astra Control Center eine URL verfügbar gemacht wird.

Die Einstellungsschritte unterscheiden sich je nach Typ des Ingress-Controllers. Astra Control Center unterstützt viele Ingress-Controller-Typen. Diese Einrichtungsverfahren bieten Beispielschritte für einige gängige Typen von Ingress-Controllern.

Bevor Sie beginnen

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

Schritte für Istio Ingress

1. Konfigurieren Sie Istio Ingress.

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

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

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

   Beispielbefehl:

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

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

   Beispielbefehl:

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

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

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

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Übernehmen Sie die Änderungen:

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

6. Überprüfen Sie den Status des Eingangs:

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

Antwort:

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

7. Astra Control Center-Installation abschließen.

Schritte für Nginx Ingress Controller

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

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

3. Übernehmen Sie die Änderungen:

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

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

Schritte für OpenShift-Eingangs-Controller

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

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Melden Sie sich in der UI des Astra Control Center an

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

Schritte

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

   Sie können nach der Anmeldung ein benutzerdefiniertes Zertifikat erstellen.

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

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

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

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

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

Beheben Sie die Fehlerbehebung für die Installation

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

Optionen

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

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

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

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

Alternative Installationsverfahren

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

Wie es weiter geht

- (Optional) Verarbeiten Sie abhängig von Ihrer Umgebung nach der Installation vollständig "Konfigurationsschritte".
- "Nachdem Sie Astra Control Center installiert haben, sich bei der UI einloggen und Ihr Passwort ändern, sollten Sie eine Lizenz einrichten, Cluster hinzufügen, die Authentifizierung aktivieren, den Storage managen und Buckets hinzufügen".

Konfigurieren Sie einen externen Zertifikaten-Manager

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

Schritte

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

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

   Beispielantwort:

   ```
   cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
   Running    0      6d5h
   cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
   Running    0      6d5h
   cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
   Running    0      6d5h
   ```

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

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

   Beispielantwort:

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

3. Erstellen eines Geheimnisses mit zuvor generierten Dateien:

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

Beispielantwort:

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

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

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Beispielantwort:

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

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

```
kubectl get ClusterIssuer
```

Beispielantwort:

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

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

Installieren Sie Astra Control Center mit OpenShift OperatorHub

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

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

Bevor Sie beginnen

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

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

- **Gewährleistung einer gesunden Clusteroperatoren und API-Dienste**:
  - Stellen Sie in Ihrem OpenShift-Cluster sicher, dass sich alle Clusterbetreiber in einem ordnungsgemäßen Zustand befinden:

```
oc get clusteroperators
```

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

```
oc get apiservices
```

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

- **Kubernetes-Ingress-Controller einrichten**: Wenn Sie über einen Kubernetes-Ingress-Controller verfügen, der den externen Zugriff auf Services wie Lastverteilung in einem Cluster verwaltet, müssen Sie ihn für die Verwendung mit Astra Control Center einrichten:

  a. Erstellen Sie den Operator-Namespace:

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

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

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

Sie sollten auch Folgendes berücksichtigen:

- **Zugriff auf die NetApp Astra Control Image Registry**:

  Sie haben die Möglichkeit, Installations-Images und Funktionserweiterungen für Astra Control, wie z. B. Astra Control Provisioner, aus der NetApp-Image-Registrierung zu beziehen.

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

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

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

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

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

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

  Die Integration von Astra Control Center mit einem Service-Mesh ist nur während des Astra Control Center möglich "Installation" Und nicht unabhängig von diesem Prozess. Ein Wechsel von einer vernetzten in eine nicht vernetzte Umgebung wird nicht unterstützt.

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

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

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Schritte

- Laden Sie das Astra Control Center herunter und extrahieren Sie es
- wenn Sie eine lokale Registrierung verwenden
- Suchen Sie die Installationsseite des Bedieners
- Installieren Sie den Operator
- Installieren Sie Astra Control Center

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

Laden Sie das Astra Control Center herunter und extrahieren Sie es

Laden Sie die Bilder des Astra Control Center von einem der folgenden Standorte herunter:

- **Astra Control Service Image Registry**: Verwenden Sie diese Option, wenn Sie keine lokale Registrierung mit den Astra Control Center Images verwenden oder wenn Sie diese Methode dem Bundle-Download von der NetApp Support-Website vorziehen.
- **NetApp Support-Seite**: Verwenden Sie diese Option, wenn Sie eine lokale Registrierung mit den Astra Control Center-Images verwenden.

Astra Control-Image-Registrierung

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

NetApp Support Website

1. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`) Vom "Download-Seite für Astra Control Center".
2. (Empfohlen, aber optional) Laden Sie das Zertifikaten- und Unterschriftenpaket für Astra Control Center herunter (`astra-control-center-certs-[version].tar.gz`) Um die Signatur des Bündels zu überprüfen.

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

```
openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
-signature certs/astra-control-center-[version].tar.gz.sig astra-
control-center-[version].tar.gz
```

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

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

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

Führen Sie zusätzliche Schritte durch, wenn Sie eine lokale Registrierung verwenden

Wenn Sie planen, das Astra Control Center Bundle in Ihre lokale Registry zu schieben, müssen Sie das NetApp Astra kubectl Kommandozeilen-Plugin verwenden.

Installieren Sie das NetApp Astra kubectl Plug-in

Führen Sie diese Schritte aus, um das neueste NetApp Astra kubectl Kommandozeilen-Plugin zu installieren.

Bevor Sie beginnen

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

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

Schritte

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

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

```
ls kubectl-astra/
```

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

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

Fügen Sie die Bilder zu Ihrer Registrierung hinzu

1. Wenn Sie planen, das Astra Control Center-Paket in Ihre lokale Registrierung zu übertragen, führen Sie die entsprechende Schrittfolge für Ihre Container-Engine aus:

Docker

a. Wechseln Sie in das Stammverzeichnis des Tarballs. Sie sollten den sehen `acc.manifest.bundle.yaml` Datei und diese Verzeichnisse:

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

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

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

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

Podman

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

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

b. Melden Sie sich bei Ihrer Registrierung an:

```
podman login <YOUR_REGISTRY>
```

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

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

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

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Telefonbuch ändern:

```
cd manifests
```

Suchen Sie die Installationsseite des Bedieners

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

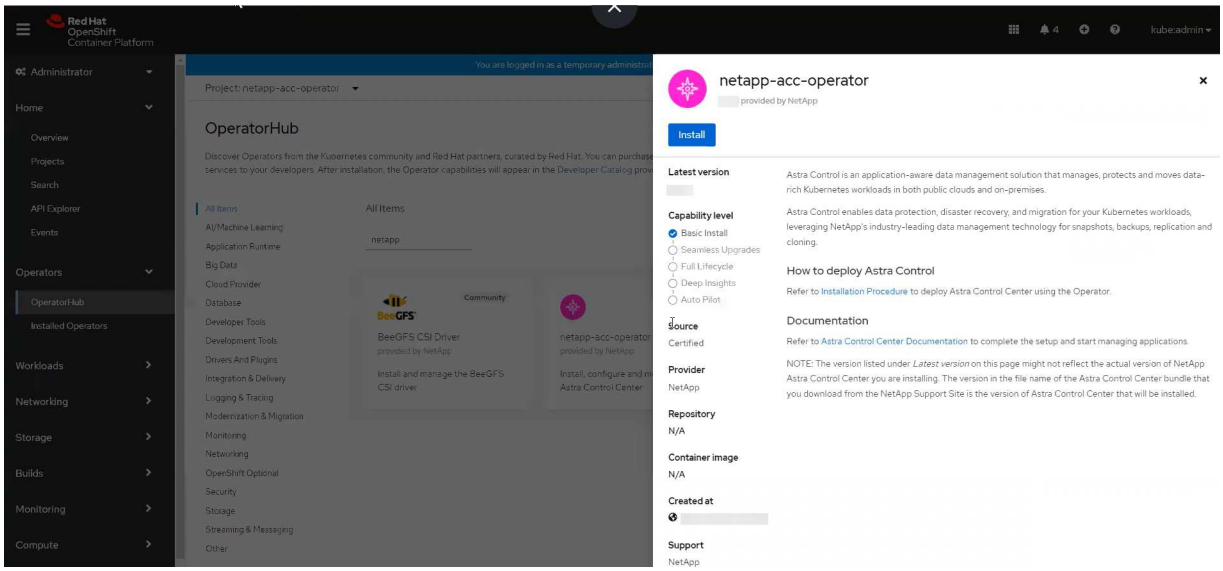
Red hat OpenShift -Webkonsole

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

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

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

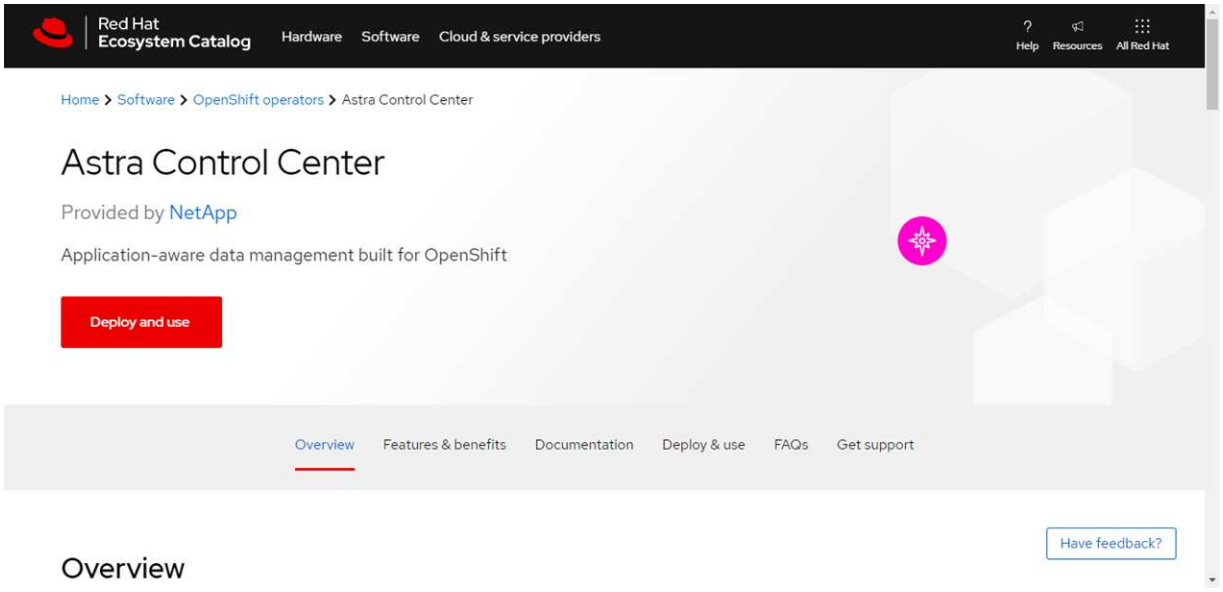
c. Suche nach `netapp-acc` Und wählen Sie den Fahrer des NetApp Astra Control Centers aus.



Red Hat Ecosystem Catalog

a. Wählen Sie das NetApp Astra Control Center aus "Operator".

b. Wählen Sie **Deploy and use**.



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.

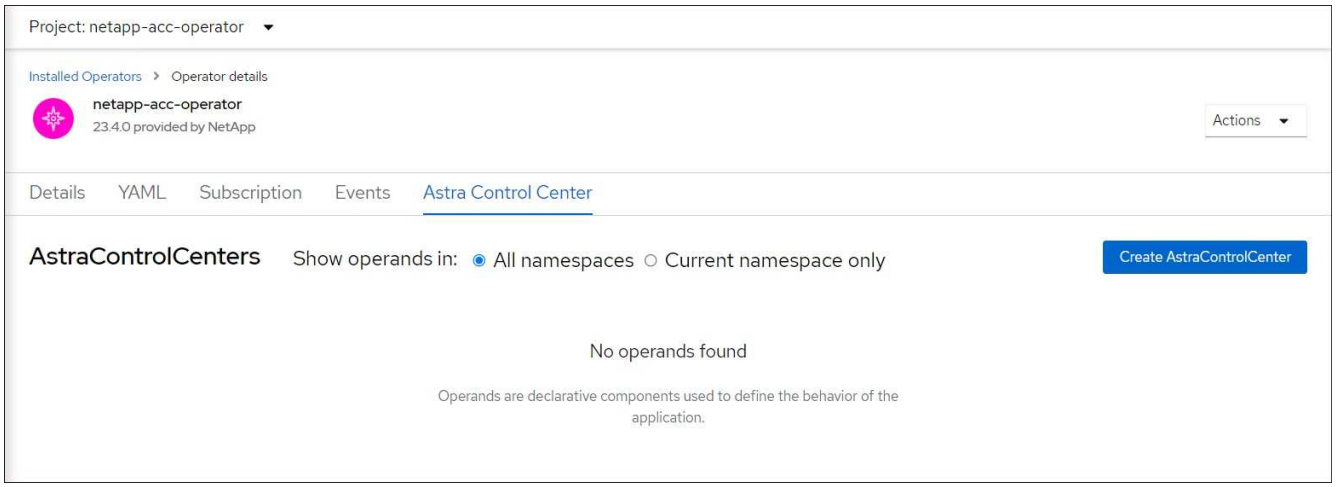
   c. Wählen Sie **Installieren**.

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

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

Installieren Sie Astra Control Center

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



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

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

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

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

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

   e. Geben Sie die Astra Control Center-Version ein, z. B. 24.02.0-69.

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

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

h. Wählen Sie die Skalierungsgröße der Installation aus.

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

i. Wählen Sie den Typ der Eindringen aus:

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

  Verwenden Sie diese Option, wenn Sie einen anderen Ingress-Controller verwenden oder Ihren eigenen Ingress-Controller verwenden möchten. Nachdem Astra Control Center bereitgestellt ist, müssen Sie das konfigurieren "Eingangs-Controller" Um Astra Control Center mit einer URL zu zeigen.

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

  Verwenden Sie diese Option, wenn Sie keinen Ingress-Controller konfigurieren möchten. Dies implementiert das Astra Control Center `traefik` Gateway als Service vom Typ Kubernetes „Load Balancer“.

Astra Control Center nutzt einen Service vom Typ „loadbalancer“ (`svc/traefik` Im Astra Control Center Namespace) und erfordert, dass ihm eine zugängliche externe IP-Adresse zugewiesen wird. Wenn in Ihrer Umgebung Load Balancer zugelassen sind und Sie noch keine konfiguriert haben, können Sie MetalLB oder einen anderen externen Service Load Balancer verwenden, um dem Dienst eine externe IP-Adresse zuzuweisen. In der Konfiguration des internen DNS-Servers sollten Sie den ausgewählten DNS-Namen für Astra Control Center auf die Load-Balanced IP-Adresse verweisen.

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

a. Verwenden Sie in **Image Registry** den Standardwert, es sei denn, Sie haben eine lokale Registrierung konfiguriert. Ersetzen Sie bei einer lokalen Registrierung diesen Wert durch den lokalen Pfad für die Bildregistrierung, in den Sie die Bilder in einem vorherigen Schritt verschoben haben. Kommen Sie nicht herein `http://` Oder `https://` Im Adressfeld.

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

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

c. Geben Sie den Vornamen des Administrators ein.

d. Konfiguration der Ressourcenskalierung

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

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

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

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

Erstellen Sie einen Registrierungsschlüssel

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

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

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

2. Erstellen eines Geheimnisses in diesem Namespace:

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

Astra Control unterstützt nur die Geheimnisse der Docker-Registrierung.

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

Wie es weiter geht

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

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

Mit Astra Control Center können Sie Ihre Applikationen in einer Hybrid-Cloud-Umgebung mit automatisierten Kubernetes-Clustern und Cloud Volumes ONTAP Instanzen managen. Sie können Astra Control Center in Ihren lokalen Kubernetes-Clustern oder in einem der automatisierten Kubernetes-Cluster in der Cloud-Umgebung bereitstellen.

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

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

folgenden Schritte aus.

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

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

Implementieren Sie Astra Control Center in Amazon Web Services

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

Was Sie für AWS benötigen

Vor der Implementierung von Astra Control Center in AWS müssen Sie folgende Aspekte beachten:

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

Anforderungen der Betriebsumgebung für AWS

Astra Control Center erfordert die folgende Betriebsumgebung für AWS:

- Red hat OpenShift Container Platform 4.11 bis 4.13

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

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung bestimmte Ressourcen. Siehe "Anforderungen an die Betriebsumgebung des Astra Control Centers".

> Das AWS Registrierungs-Token läuft in 12 Stunden ab. Danach müssen Sie den Schlüssel für die Docker Image-Registrierung erneuern.

Überblick über die Implementierung für AWS

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

Jeder dieser Schritte wird unten im Detail erklärt.

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

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

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

Siehe "Erste AWS Zugangsdaten".

Installation eines RedHat OpenShift-Clusters in AWS

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

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

Konfigurieren von AWS

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

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

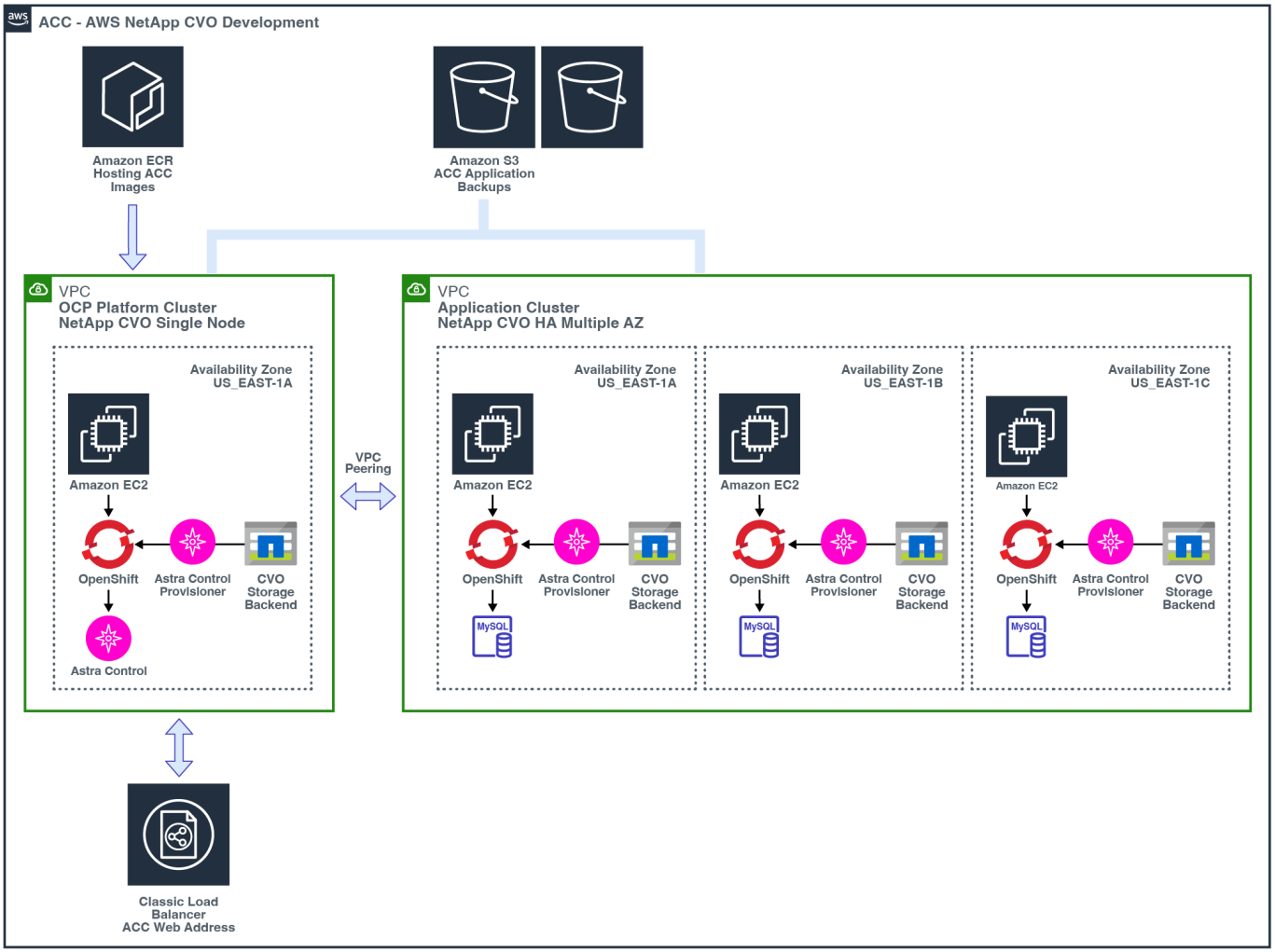
1. Virtuelles AWS Netzwerk erstellen.
2. EC2 Computing-Instanzen prüfen. Dabei können es sich um einen Bare Metal Server oder VMs in AWS handelt.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in AWS, um die Astra-Anforderungen zu erfüllen. Siehe "Anforderungen des Astra Control Centers".
4. Erstellen Sie mindestens einen AWS S3-Bucket zum Speichern Ihrer Backups.
5. (Optional) Wenn Sie nicht auf die NetApp-Image-Registrierung zugreifen können, gehen Sie wie folgt vor:
   a. Eine AWS Elastic Container Registry (ECR) erstellen, um alle Astra Control Center Images zu hosten.

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

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

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

Beispiel für eine AWS Implementierung:



Konfiguration von NetApp BlueXP für AWS

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

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

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

Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu.
2. Erstellen Sie einen Arbeitsbereich.

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

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

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

Installieren Sie Astra Control Center für AWS

Dem Standard folgen "Installationsanweisungen für Astra Control Center".

AWS verwendet den Bucket-Typ generischer S3.

Implementieren Sie Astra Control Center in der Google Cloud Platform

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

Was wird für GCP benötigt

Bevor Sie Astra Control Center in GCP implementieren, benötigen Sie folgende Elemente:

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

Anforderungen der Betriebsumgebung für GCP

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

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung bestimmte Ressourcen. Siehe "Anforderungen an die Betriebsumgebung des Astra Control Centers".

Übersicht über die Implementierung für GCP

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

Jeder dieser Schritte wird unten im Detail erklärt.

1. Installation eines RedHat OpenShift-Clusters in GCP.
2. Erstellung eines GCP-Projekts und einer virtuellen Private Cloud.
3. dass Sie über ausreichende IAM-Berechtigungen verfügen.
4. GCP konfigurieren.
5. Konfiguration von NetApp BlueXP für GCP.
6. Installieren Sie Astra Control Center für GCP.

Installation eines RedHat OpenShift-Clusters in GCP

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

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

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

Erstellung eines GCP-Projekts und einer virtuellen Private Cloud

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

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

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

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

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

Siehe "Erste GCP-Zugangsdaten und -Berechtigungen".

GCP konfigurieren

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

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

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

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

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

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

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

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

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

7. Richten Sie DNS-Zonen ein.

Konfiguration von NetApp BlueXP für GCP

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

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

Bevor Sie beginnen

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

Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu. Siehe "GCP-Konten hinzufügen".
2. Fügen Sie einen Connector für GCP hinzu.
   a. Entscheiden Sie sich für „GCP“ als Provider.
   b. GCP-Zugangsdaten eingeben. Siehe "Erstellen eines Connectors in GCP von BlueXP".
   c. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.
3. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
   a. Speicherort: „GCP“
   b. Typ: „Cloud Volumes ONTAP HA“
4. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.
   a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s** > **Cluster list** > **Cluster-Details** wählen.
   b. Beachten Sie in der oberen rechten Ecke die Version von Astra Control Provisioner.
   c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen mit „NetApp“ als provisionierung.

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

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

Installieren Sie Astra Control Center für GCP

Dem Standard folgen "Installationsanweisungen für Astra Control Center".

GCP verwendet den allgemeinen S3-Bucket-Typ.

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

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

Implementieren Sie Astra Control Center in Microsoft Azure

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

Was Sie für Azure benötigen

Vor der Implementierung von Astra Control Center in Azure benötigen Sie folgende Elemente:

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

Anforderungen an die Betriebsumgebung für Azure

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

Astra Control Center erfordert zusätzlich zu den Ressourcenanforderungen der Umgebung bestimmte Ressourcen. Siehe "Anforderungen an die Betriebsumgebung des Astra Control Centers".

Überblick über die Implementierung für Azure

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

Jeder dieser Schritte wird unten im Detail erklärt.

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

Installieren Sie einen RedHat OpenShift-Cluster auf Azure

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

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

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

Erstellen von Azure Ressourcengruppen

Erstellen Sie mindestens eine Azure-Ressourcengruppe.

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

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

Ressourcengruppe erstellen.

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

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

Siehe "Azure Zugangsdaten und Berechtigungen".

Konfigurieren Sie Azure

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

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

1. Virtuelles Azure Netzwerk erstellen.
2. Prüfen Sie die Computing-Instanzen. Dabei können es sich um einen Bare Metal Server oder VMs in Azure handelt.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in Azure, um die Astra-Anforderungen zu erfüllen. Siehe "Anforderungen des Astra Control Centers".
4. Erstellen Sie mindestens einen Azure Blob Container, um Ihre Backups zu speichern.
5. Erstellen Sie ein Speicherkonto. Sie benötigen ein Storage-Konto, um einen Container zu erstellen, der als Bucket in Astra Control Center verwendet werden soll.
6. Erstellen eines Geheimnisses, das für den Bucket-Zugriff erforderlich ist
7. (Optional) Wenn Sie nicht auf die NetApp-Image-Registrierung zugreifen können, gehen Sie wie folgt vor:

    a. Azure Container Registry (ACR) erstellen, um die Astra Control Center Images zu hosten.
    b. ACR-Zugriff für Docker Push/Pull für alle Astra Control Center Images einrichten
    c. Übertragen Sie die Astra Control Center-Images mithilfe des folgenden Skripts in diese Registrierung:

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

    - Beispiel*:

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

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

8. Richten Sie DNS-Zonen ein.

Konfiguration von NetApp BlueXP (ehemals Cloud Manager) für Azure

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

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

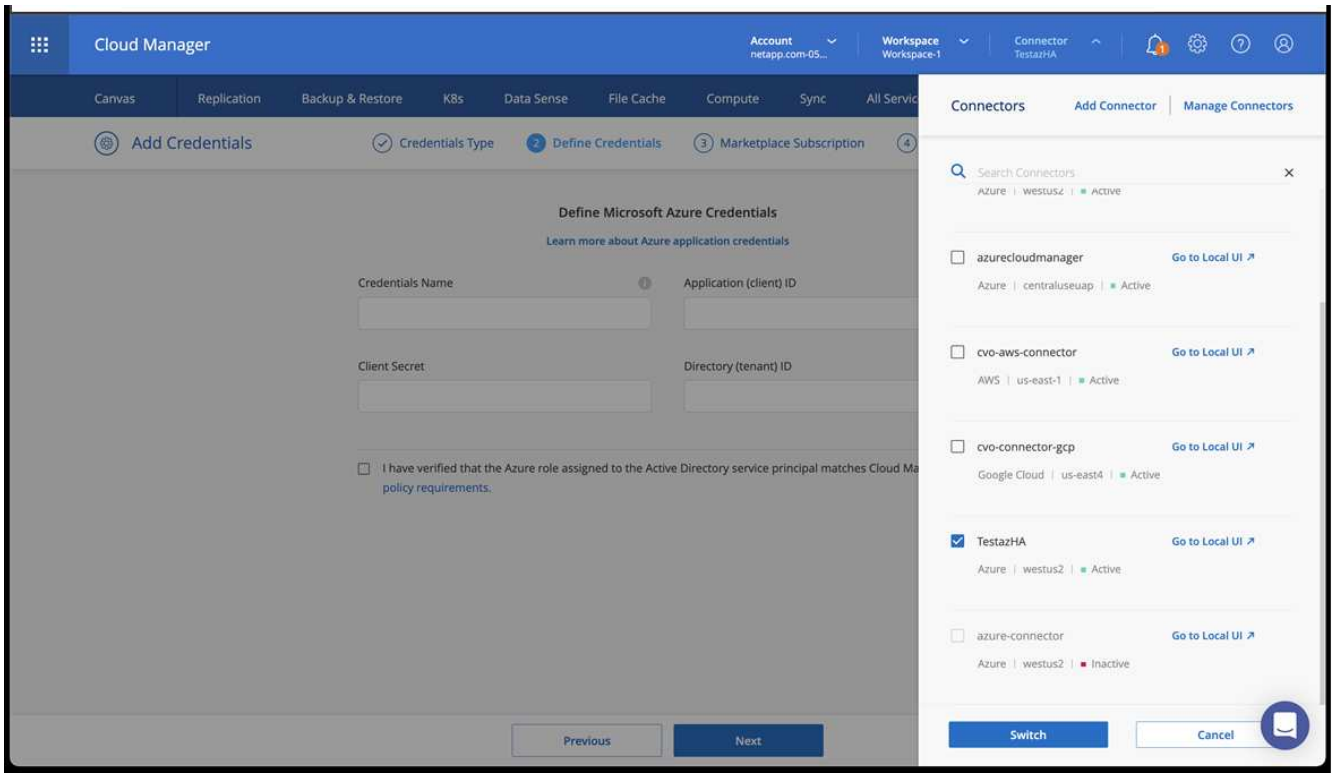
Bevor Sie beginnen

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

Schritte

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

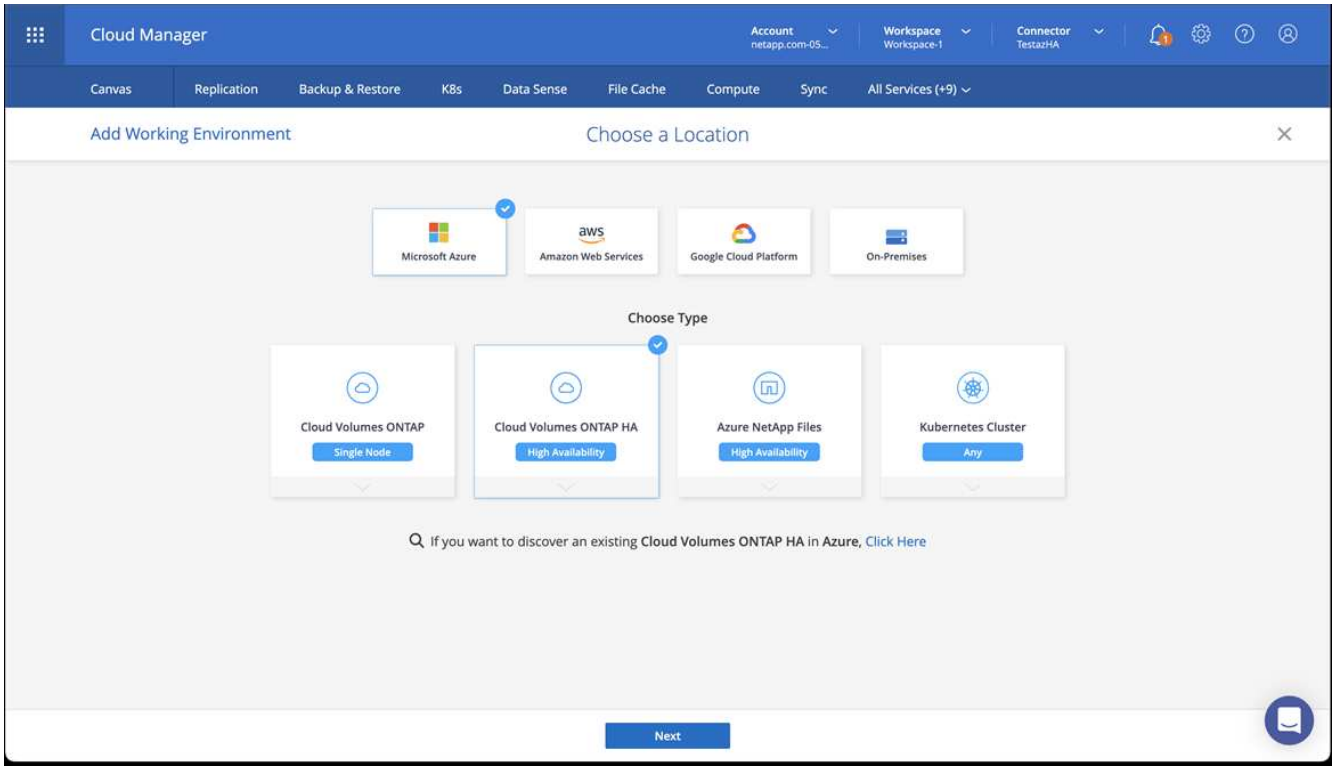
      Siehe "Erstellen eines Konnektors in Azure aus BlueXPr".
3. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.



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

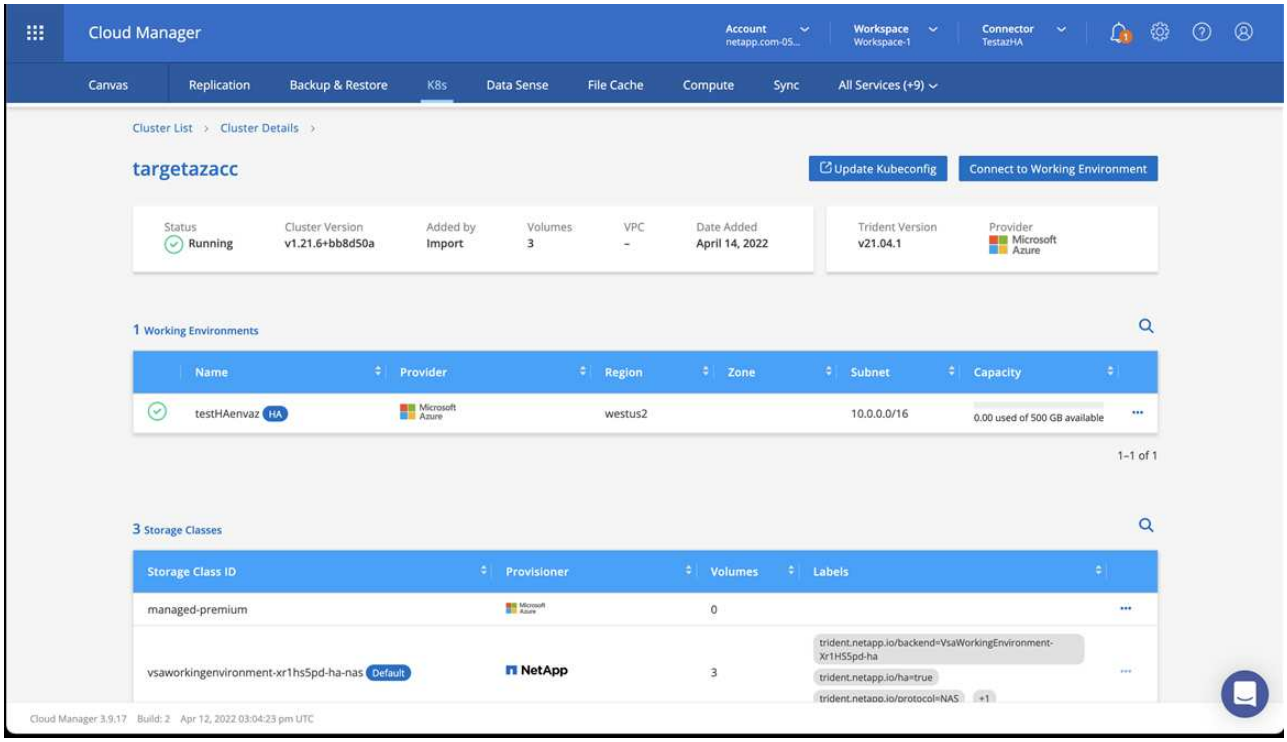
   a. Ort: „Microsoft Azure“.

   b. Typ: „Cloud Volumes ONTAP HA“.



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

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



b. Beachten Sie in der oberen rechten Ecke die Version von Astra Control Provisioner.

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

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

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

Installation und Konfiguration von Astra Control Center für Azure

Installieren Sie Astra Control Center standardmäßig "Installationsanweisungen".

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

Konfigurieren Sie nach der Installation das Astra Control Center

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

Ressourceneinschränkungen entfernen

In einigen Umgebungen werden die Objekte ResourceQuotas und LimitRanges verwendet, um zu verhindern, dass die Ressourcen in einem Namespace alle verfügbaren CPUs und Speicher im Cluster verbrauchen. Das Astra Control Center stellt keine Höchstgrenzen ein, sodass diese Ressourcen nicht eingehalten werden.

Wenn Ihre Umgebung auf diese Weise konfiguriert ist, müssen Sie diese Ressourcen aus den Namespaces entfernen, in denen Sie Astra Control Center installieren möchten.

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

Schritte

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

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

Antwort:

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

2. Alle Ressourcen-Kontingente nach Namen löschen:

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

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

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

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

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

Antwort:

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

4. Grenzwerte nach Namen löschen:

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

Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

Astra Control Center verwendet standardmäßig ein selbstsigniertes TLS-Zertifikat für Ingress-Controller-Datenverkehr (nur in bestimmten Konfigurationen) und die Web-UI-Authentifizierung mit Webbrowsern. Für den Produktionsbetrieb sollten Sie das vorhandene selbstsignierte TLS-Zertifikat entfernen und durch ein von einer Zertifizierungsstelle signiertes TLS-Zertifikat ersetzen.

Das selbstsignierte Standardzertifikat wird für zwei Verbindungstypen verwendet:

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

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

Bevor Sie beginnen

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

Entfernen Sie das selbstsignierte Zertifikat

Entfernen Sie das vorhandene selbstsignierte TLS-Zertifikat.

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

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

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

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

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

Fügen Sie mithilfe der Befehlszeile ein neues Zertifikat hinzu

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

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

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

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

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

   CRD:

```
#spec:
#  selfSigned: {}

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

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

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

   Antwort:

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

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

> In diesem Beispiel wird der verwendet `dnsNames` Eigenschaft zur Angabe der Astra Control Center DNS-Adresse. Astra Control Center unterstützt nicht die Verwendung der CN-Eigenschaft (Common Name) zur Angabe der DNS-Adresse.

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

5. Erstellen Sie das Zertifikat mit dem folgenden Befehl:

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

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

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

Antwort:

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

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

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

CRD:

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

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

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

CRD:

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

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

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:

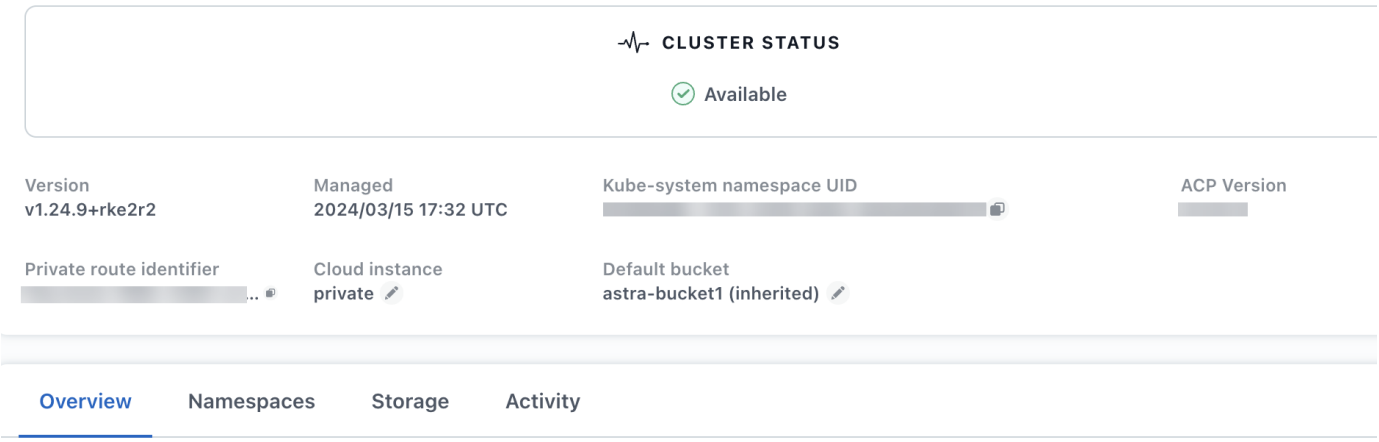
```
./tridentctl uninstall -n trident
```

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>
```

```
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>
```

```
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>
```

```
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>
```

```
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>
```

```
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
```

```
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".

Copyright-Informationen

Copyright © 2025 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

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

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

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

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

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

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

Markeninformationen

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