



# **Los geht's**

## **Astra Control Center**

NetApp  
November 21, 2023

# Inhalt

- Los geht's ..... 1
- Weitere Informationen zu Astra Control ..... 1
- Anforderungen des Astra Control Centers ..... 4
- Schnellstart für Astra Control Center ..... 9
- Übersicht über die Installation ..... 11
- Einrichten des Astra Control Center ..... 79
- Häufig gestellte Fragen zum Astra Control Center ..... 100

# 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 in Umgebungen mit mehreren Cloud-Providern und einem NetApp Cloud Volumes ONTAP Storage-Back-End installiert werden.

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

	Astra Control Service	Astra Control Center
Welche Storage-Back-Ends werden unterstützt?	<ul style="list-style-type: none"> <li>• Amazon Web Services: <ul style="list-style-type: none"> <li>◦ Amazon EBS</li> <li>◦ Amazon FSX für NetApp ONTAP</li> <li>◦ "Cloud Volumes ONTAP"</li> </ul> </li> <li>• Google Cloud: <ul style="list-style-type: none"> <li>◦ Google Persistent Disk</li> <li>◦ NetApp Cloud Volumes Service</li> <li>◦ "Cloud Volumes ONTAP"</li> </ul> </li> <li>• Microsoft Azure: <ul style="list-style-type: none"> <li>◦ Über Azure Gemanagte Festplatten</li> <li>◦ Azure NetApp Dateien</li> <li>◦ "Cloud Volumes ONTAP"</li> </ul> </li> <li>• Self-Managed Cluster: <ul style="list-style-type: none"> <li>◦ Amazon EBS</li> <li>◦ Google Persistent Disk</li> <li>◦ Über Azure Gemanagte Festplatten</li> <li>◦ "Cloud Volumes ONTAP"</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• NetApp ONTAP AFF und FAS Systeme</li> <li>• "Cloud Volumes ONTAP"</li> </ul>

## 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.
- Verwendet diese neue Administratorrolle für die Installation "[Astra Trident](#)" Auf dem Cluster und um eine oder mehrere Storage-Klassen zu erstellen.
- Wenn Sie ein Cloud-Service-Storage-Angebot von NetApp als Storage-Back-End verwenden, verwendet der Astra Control Service Astra Trident zur Bereitstellung persistenter Volumes für Ihre Applikationen. 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 Astra Trident-basierter Storage-Klasse mit einem Storage-Back-End von ONTAP 9.5 und höher.

In einer Cloud-vernetzten Umgebung nutzt Astra Control Center erweiterte Monitoring- und Telemetriedaten mithilfe von Cloud Insights. Liegt keine Cloud Insights-Verbindung vor, ist das Monitoring und die Telemetrie nur begrenzt (7 Tage Metriken) im Astra Control Center verfügbar und wird auch über offene Messpunkte in native Kubernetes-Monitoring-Tools (wie Prometheus und Grafana) exportiert.

Astra Control Center ist vollständig in das AutoSupport und Active IQ Ecosystem integriert, damit Benutzer und NetApp Support Fehlerbehebungs- und Verwendungsinformationen liefern können.

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"](#)
- ["Verwenden Sie die Astra Control API"](#)
- ["Cloud Insights-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](#)
- [Ressourcenanforderungen des Host-Clusters](#)
- [Anforderungen von Astra Trident](#)
- [Storage-Back-Ends](#)
- [Bildregistrierung](#)
- [Astra Control Center-Lizenz](#)
- [ONTAP-Lizenzen](#)
- [Netzwerkanforderungen](#)
- [Ingress für lokale Kubernetes Cluster](#)
- [Unterstützte Webbrowser](#)
- [Zusätzliche Anforderungen an Applikations-Cluster](#)

## 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 1.23 und 1.24
Google Anthos	1.12 bis 1.14 (siehe <a href="#">Anforderungen für Google Anthos Ingress</a> )
Kubernetes (Vorgelagert)	1.24 bis 1.26 (für Kubernetes 1.25 oder neuer ist Astra Trident 22.10 oder neuer erforderlich)
Rancher Kubernetes Engine (RKE)	RKE 1.3 mit Rancher 2.6 RKE 1.4 mit Rancher 2.7 RKE 2 (v1.23.x) mit Rancher 2.6 RKE 2 (v1.24.x) mit Rancher 2.7
Red hat OpenShift Container Platform	4.10 bis 4.12
VMware Tanzu Kubernetes Grid	1.6 (Siehe <a href="#">Ressourcenanforderungen des Host-Clusters</a> )
VMware Tanzu Kubernetes Grid Integrated Edition	1.14 und 1.15 (siehe <a href="#">Ressourcenanforderungen des Host-Clusters</a> )

## 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.
  - Achten Sie bei der Implementierung des Astra Control Center in einer TKG- oder TKGi-Umgebung auf

die speziellen Anforderungen von Astra Trident. Weitere Informationen finden Sie im "[Astra Trident-Dokumentation](#)".

## Anforderungen von Astra Trident

Stellen Sie sicher, dass Sie die folgenden Anforderungen für Astra Trident erfüllen, die Ihren Anforderungen Ihrer Umgebung entsprechen:

- **Mindestversion für Astra Control Center:** Astra Trident 22.04 oder neuer installiert und konfiguriert.
- **SnapMirror Replizierung:** Installation von Astra Trident 22.07 oder neuer für die SnapMirror-basierte Applikationsreplizierung
- **Für die Unterstützung von Kubernetes 1.25 oder neuer:** Astra Trident 22.10 oder höher für Kubernetes 1.25 oder neuere Cluster installiert (Sie müssen vor dem Upgrade auf Kubernetes 1.25 oder neuer auf Astra Trident 22.10 upgraden)
- **ONTAP-Konfiguration mit Astra Trident:**
  - **Storage class:** Konfigurieren Sie mindestens eine Astra Trident 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 die Workerknoten in Ihrem Cluster mit den entsprechenden Speichertreibern konfiguriert sind, 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` (App-Replizierung ist bei diesem Storage-Klassen-Typ nicht verfügbar)
    - `ontap-nas-economy` (Snapshots, Replizierungsrichtlinien und Sicherungsrichtlinien stehen bei diesem Typ von Storage-Klassen nicht zur Verfügung.)

## Storage-Back-Ends

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

- **Unterstützte Backends:** Astra Control Center unterstützt folgende Speicher-Backends:
  - NetApp ONTAP 9.8 oder neuer AFF, FAS und ASA Systeme
  - NetApp ONTAP Select 9.8 oder höher
  - NetApp Cloud Volumes ONTAP 9.8 oder höher
- **Erforderliche Back-End-Speicherkapazität:** Mindestens 500 GB 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"](#).

## Bildregistrierung

Sie müssen über eine vorhandene private Docker Image-Registrierung verfügen, auf die Sie Astra Control Center Build-Images übertragen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen.

## 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 nicht, sind einige Funktionen möglicherweise begrenzt, beispielsweise das Empfangen von Monitoring- und Kennzahlendaten von NetApp Cloud Insights oder das Senden von Support-Paketen an die ["NetApp Support Website"](#).
- **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 auf beiden Wegen zwischen dem Cluster geöffnet ist, der Astra Control Center hostet, und jedem verwalteten Cluster
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)
Astra Control Center	Gehosteter Cloud Insights Service ( <a href="https://www.netapp.com/cloud-services/cloud-insights/">https://www.netapp.com/cloud-services/cloud-insights/</a> )	443	HTTPS	Cloud Insights Kommunikation
Astra Control Center	Amazon S3 Storage-Bucket-Provider	443	HTTPS	Amazon S3 Storage-Kommunikation
Astra Control Center	NetApp AutoSupport ( <a href="https://support.netapp.com">https://support.netapp.com</a> )	443	HTTPS	Kommunikation zwischen NetApp AutoSupport

## 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 ["Konfigurieren Sie 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 für 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 die von Astra Trident unterstützte Version enthält"](#)
- ["Bereiten Sie die Worker-Knoten vor"](#)
- ["Konfigurieren Sie das Astra Trident Storage-Back-End"](#)
- ["Konfigurieren Sie Astra Trident Storage-Kurse"](#)
- ["Installieren Sie den Astra Trident Volume Snapshot Controller"](#)
- ["Erstellen Sie eine Volume Snapshot-Klasse"](#)

## ONTAP-Anmeldedaten

- ["Konfigurieren Sie die ONTAP-Anmeldedaten"](#)

2

### 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:

- ["Applikationsmanagement"](#): Ressourcen definieren, die verwaltet werden sollen.
- ["Schützen von Applikationen"](#): Schutzrichtlinien konfigurieren und Anwendungen replizieren, klonen und migrieren.
- ["Konten verwalten"](#): Benutzer, Rollen, LDAP, Anmeldeinformationen und mehr.
- ["Optional Verbindung mit Cloud Insights herstellen"](#): Anzeige von Kennzahlen zur Gesundheit Ihres Systems.

## Finden Sie weitere Informationen

- ["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

Laden Sie zum Installieren des Astra Control Center das Installationspaket von der NetApp Support Site herunter und führen Sie die folgenden Schritte aus. Mit diesem Verfahren können Sie Astra Control Center in Internet-angeschlossenen oder luftgekapselten Umgebungen installieren.

### Andere Installationsverfahren

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

Eine Demonstration des Installationsvorgangs für Astra Control Center finden Sie unter ["Dieses Video"](#).

### Bevor Sie beginnen

- ["Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor"](#).
- Wenn Sie POD-Sicherheitsrichtlinien in Ihrer Umgebung konfiguriert haben oder konfigurieren möchten, sollten Sie sich mit den POD-Sicherheitsrichtlinien vertraut machen und wie diese sich auf die Installation des Astra Control Center auswirken. Siehe ["Einschränkungen der POD-Sicherheitsrichtlinie verstehen"](#).
- Stellen Sie sicher, dass alle API-Services in einem ordnungsgemäßen Zustand und verfügbar sind:

```
kubectl get apiservices
```

- Stellen Sie sicher, dass der Astra FQDN, den Sie verwenden möchten, für diesen Cluster routingfähig ist. Das bedeutet, dass Sie entweder einen DNS-Eintrag in Ihrem internen DNS-Server haben oder eine bereits registrierte Core URL-Route verwenden.
- Wenn bereits ein Zertifikat-Manager im Cluster vorhanden ist, müssen Sie einen Teil 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.



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

### Über diese Aufgabe

Mit dem Astra Control Center-Installationsprozess können Sie Folgendes tun:

- Installieren Sie die Astra-Komponenten im `netapp-acc` (Oder Name des benutzerdefinierten Namespace).
- Erstellen Sie ein Standard-Astra Control Owner-Administratorkonto.
- Legen Sie eine E-Mail-Adresse für einen Administrator und ein Standard-Kennwort für die Ersteinrichtung fest. Diesem Benutzer wird die Owner-Rolle zugewiesen, die für die erste Anmeldung bei der UI benötigt wird.
- Stellen Sie fest, dass alle Astra Control Center Pods ausgeführt werden.
- Installieren Sie die Astra Control Center-UI.



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.

### Schritte

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

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

### Laden Sie das Astra Control Center herunter und extrahieren Sie es

1. Wechseln Sie zum "[Download-Seite für Astra Control Center](#)" Auf der NetApp Support Site
2. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`).
3. (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 Pakets 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.

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

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

### Installieren Sie das NetApp Astra kubect1 Plug-in

Sie können das NetApp Astra kubect1 Befehlszeilenschnittstelle-Plug-in verwenden, um Images in ein lokales Docker Repository zu verschieben.

#### Bevor Sie beginnen

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

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

#### Schritte

1. Geben Sie die verfügbaren Plug-ins-Binärdateien von NetApp Astra kubect1 an und notieren Sie sich den Namen der für Ihr Betriebssystem und die CPU-Architektur erforderlichen Datei:



Die kubect1 Plugin-Bibliothek ist Teil des tar-Bündels und wird in den Ordner extrahiert `kubect1-astra`.

```
ls kubect1-astra/
```

2. Verschieben Sie die richtige Binärdatei in den aktuellen Pfad, und benennen Sie sie in um `kubect1-astra`:

```
cp kubect1-astra/<binary-name> /usr/local/bin/kubect1-astra
```

### Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

1. Führen Sie die entsprechende Schrittfolge für Ihre Container-Engine durch:

## Docker

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

```
acc.manifest.bundle.yaml
acc/
```

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

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

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

## Podman

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

```
acc.manifest.bundle.yaml
acc/
```

2. Melden Sie sich bei Ihrer Registrierung an:

```
podman login <YOUR_REGISTRY>
```

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

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



```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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

```

**Podman 3**

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/23.04.2-7/image:version>

## Einrichten von Namespace und Geheimdienstraum für Registrys mit auth Anforderungen

1. Exportieren Sie den KUBECONFIG für den Hostcluster Astra Control Center:

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



Bevor Sie die Installation abgeschlossen haben, stellen Sie sicher, dass Ihr KUBECONFIG auf den Cluster zeigt, in dem Sie Astra Control Center installieren möchten. Die KUBECONFIG kann nur einen Kontext enthalten.

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
```

Antwort:

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

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



Platzhalter `your_registry_path` Sollte die Position der Bilder, die Sie früher hochgeladen haben, entsprechen (z. B. `[Registry_URL]/netapp/astra/astracc/23.04.2-7`).

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

Beispielantwort:

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



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

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

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

Beispielantwort:

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

- d. Erstellen Sie ein Geheimnis für das netapp-acc (Oder Name des benutzerdefinierten Namespace). Fügen Sie Docker-Informationen hinzu und führen Sie den folgenden Befehl aus:

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

Antwort

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

## Installieren Sie den Operator Astra Control Center

1. Telefonbuch ändern:

```
cd manifests
```

2. Bearbeiten Sie die YAML-Implementierung des Astra Control Center-Bediensers (astra\_control\_center\_operator\_deploy.yaml) Zu Ihrem lokalen Register und Geheimnis zu verweisen.

```
vim astra_control_center_operator_deploy.yaml
```



Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

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

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

- b. Ändern [your\_registry\_path] Für das kube-rbac-proxy Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben [Vorheriger Schritt](#).
- c. Ändern [your\_registry\_path] Für das acc-operator-controller-manager Bild zum Registrierungspfad, in dem Sie die Bilder in ein geschoben haben [Vorheriger Schritt](#).

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```

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: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
            - name: ACCOP_HELM_INSTALLTIMEOUT
              value: 5m
            image: [your_registry_path]/acc-operator:23.04.36
          imagePullPolicy: IfNotPresent
          livenessProbe:
            httpGet:
              path: /healthz
              port: 8081
            initialDelaySeconds: 15

```

```
    periodSeconds: 20
  name: manager
  readinessProbe:
    httpGet:
      path: /readyz
      port: 8081
    initialDelaySeconds: 5
    periodSeconds: 10
  resources:
    limits:
      cpu: 300m
      memory: 750Mi
    requests:
      cpu: 100m
      memory: 75Mi
  securityContext:
    allowPrivilegeEscalation: false
imagePullSecrets: []
  securityContext:
    runAsUser: 65532
  terminationGracePeriodSeconds: 10
```

### 3. Installieren Sie den Astra Control Center-Operator:

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

Beispielantwort:

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

#### 4. Überprüfen Sie, ob Pods ausgeführt werden:

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

### Konfigurieren Sie Astra Control Center

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

```
vim astra_control_center.yaml
```



Ein YAML-Beispiel mit Anmerkungen folgt diesen Schritten.

2. Ändern oder bestätigen Sie die folgenden Einstellungen:

`<code>accountName</code>`

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

`<code>astraVersion</code>`

Einstellung	Anleitung	Typ	Beispiel
astraVersion	Die zu implementierende Version des Astra Control Center: Für diese Einstellung ist keine Aktion erforderlich, da der Wert bereits ausgefüllt wird.	Zeichenfolge	23.04.2-7

`<code>astraAddress</code>`

Einstellung	Anleitung	Typ	Beispiel
astraAddress	<p>Ändern Sie das <code>astraAddress</code> 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 <a href="#">"Anforderungen des Astra Control Centers"</a>.</p> <p>HINWEIS: Nicht verwenden <code>http://</code> Oder <code>https://</code> In der Adresse. Kopieren Sie diesen FQDN zur Verwendung in einem <a href="#">Später Schritt</a>.</p>	Zeichenfolge	astra.example.com



## <code>autoSupport</code>

Anhand Ihrer Auswahl in diesem Abschnitt wird bestimmt, ob Sie an der pro-aktiven Support-Applikation von NetApp, dem NetApp Active IQ und dem Sendeort von Daten teilnehmen. Eine Internetverbindung ist erforderlich (Port 442), und alle Supportdaten werden anonymisiert.

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

`<code>email</code>`

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

`<code>firstName</code>`

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

`<code>lastName</code>`

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

`<code>imageRegistry</code>`

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

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

`<code>storageClass</code>`

Einstellung	Anleitung	Typ	Beispiel
storageClass	<p>Ändern Sie das <code>storageClass</code> Wert von <code>ontap-gold</code> Je nach Installationsanforderungen zu einer anderen Ressource für Astra Trident Storage Class wechseln. Führen Sie den Befehl aus</p> <pre>kubectl get sc</pre> <p>So ermitteln Sie Ihre vorhandenen konfigurierten Speicherklassen. In die Manifest-Datei muss eine der Astra Trident-basierten Storage-Klassen eingegeben werden (<code>astra-control-center-&lt;version&gt;.manifest</code>) Und wird für Astra PVS verwendet. Wenn er nicht festgelegt ist, wird die Standard-Speicherklasse verwendet.</p> <p><b>HINWEIS:</b> Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass diese die einzige Storage-Klasse mit der Standardbeschriftung ist.</p>	Zeichenfolge	ontap-gold

`<code>volumeReclaimPolicy</code>`

Einstellung	Anleitung	Typ	Optionen
volumeReclaimPolicy	Damit wird die Rückgewinnungsrichtlinie für die PVS von Astra festgelegt. Festlegen dieser Richtlinie auf Retain Behält persistente Volumes nach dem Löschen von Astra bei. Festlegen dieser Richtlinie auf Delete Löscht persistente Volumes nach dem Löschen von astra. Wenn dieser Wert nicht festgelegt ist, werden die PVS beibehalten.	Zeichenfolge	<ul style="list-style-type: none"><li>• Retain (Dies ist der Standardwert)</li><li>• Delete</li></ul>

`<code>ingressType</code>`





Einstellung	Anleitung	Typ	Optionen
ingressType	<p>Verwenden Sie einen der folgenden Eingangstypen:</p> <p><b>Generic</b> (ingressType: "Generic") (Standard) Verwenden Sie diese Option, wenn Sie einen anderen Ingress-Controller verwenden oder Ihren eigenen Ingress-Controller verwenden möchten. Nach der Implementierung des Astra Control Center müssen Sie den konfigurieren <b>"Eingangs-Controller"</b> Um Astra Control Center mit einer URL zu zeigen.</p> <p><b>AccTraefik</b> (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:</p> <p>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</p>	Zeichenfolge	<ul style="list-style-type: none"> <li>• Generic (Dies ist der Standardwert)</li> <li>• AccTraefik</li> </ul>



`<code>scaleSize</code>`

Einstellung	Anleitung	Typ	Optionen
scaleSize	<p>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.</p> <p>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.</p>	Zeichenfolge	<ul style="list-style-type: none"><li>• Small</li><li>• Medium (Dies ist der Standardwert)</li></ul>

`<code>astraResourcesScaler</code>`

Einstellung	Anleitung	Typ	Optionen
<code>astraResourcesScaler</code>	<p>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.</p> <p>In Szenarien mit kleineren Entwicklungs- oder Testclustern jedoch das CR-Feld <code>astraResourcesScaler</code> Kann auf festgelegt werden <code>Off</code>. Dadurch werden Ressourcenanforderungen deaktiviert und die Bereitstellung auf kleineren Clustern ist möglich.</p>	Zeichenfolge	<ul style="list-style-type: none"><li>• Default (Dies ist der Standardwert)</li><li>• Off</li></ul>

## `<code>additionalValues</code>`

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

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

`<code>crds</code>`

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

Einstellung	Anleitung	Typ	Beispiel
<code>crds.externalCertManager</code>	<p>Wenn Sie einen externen Zertifikaten-Manager verwenden, ändern Sie <code>externalCertManager</code> Bis <code>true</code>. Der Standardwert <code>false</code> Führt dazu, dass Astra Control Center während der Installation seine eigenen CRT-Manager-CRDs installiert.</p> <p>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.</p>	Boolesch	<code>False</code> (Dieser Wert ist der Standardwert)
<code>crds.externalTraffic</code>	<p>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.</p>	Boolesch	<code>False</code> (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.

```
<strong>astra_control_center.yaml</strong>
```

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues: {}
  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]
```

Beispielantwort:

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

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

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

Beispielantwort:

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



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 `Running` haben. Es kann mehrere Minuten dauern, bis die System-Pods implementiert sind.

## Beispielantwort

NAME	READY	STATUS	
RESTARTS	AGE		
acc-helm-repo-6cc7696d8f-pmhm8	1/1	Running	0
9h			
activity-597fb656dc-5rd4l	1/1	Running	0
9h			
activity-597fb656dc-mqmcw	1/1	Running	0
9h			
api-token-authentication-62f84	1/1	Running	0
9h			
api-token-authentication-68nlf	1/1	Running	0
9h			
api-token-authentication-ztgrm	1/1	Running	0
9h			
asup-669d4ddbc4-fnmwp	1/1	Running	1
(9h ago) 9h			
authentication-78789d7549-lk686	1/1	Running	0
9h			
bucket-service-65c7d95496-24x7l	1/1	Running	3
(9h ago) 9h			
cert-manager-c9f9fbf9f-k8zq2	1/1	Running	0
9h			
cert-manager-c9f9fbf9f-qj1zm	1/1	Running	0
9h			
cert-manager-cainjector-dbbbd8447-b5q1l	1/1	Running	0
9h			
cert-manager-cainjector-dbbbd8447-p5whs	1/1	Running	0
9h			
cert-manager-webhook-6f97bb7d84-4722b	1/1	Running	0
9h			
cert-manager-webhook-6f97bb7d84-86kv5	1/1	Running	0
9h			
certificates-59d9f6f4bd-2j899	1/1	Running	0
9h			
certificates-59d9f6f4bd-9d9k6	1/1	Running	0
9h			
certificates-expiry-check-28011180--1-81kxz	0/1	Completed	0
9h			
cloud-extension-5c9c9958f8-jdhrp	1/1	Running	0
9h			
cloud-insights-service-5cdd5f7f-pp8r5	1/1	Running	0
9h			
composite-compute-66585789f4-hxn5w	1/1	Running	0
9h			

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



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

polaris-vault-2 9h	1/1	Running	0
public-metrics-76fbf9594d-zmxzw 9h	1/1	Running	0
storage-backend-metrics-7d7fbc9cb9-lmd25 9h	1/1	Running	0
storage-provider-5bdd456c4b-2fftc 9h	1/1	Running	0
task-service-87575df85-dnn2q (9h ago) 9h	1/1	Running	3
task-service-task-purge-28011720--1-q6w4r 28m	0/1	Completed	0
task-service-task-purge-28011735--1-vk6pd 13m	1/1	Running	0
telegraf-ds-2r2kw 9h	1/1	Running	0
telegraf-ds-6s9d5 9h	1/1	Running	0
telegraf-ds-96jl7 9h	1/1	Running	0
telegraf-ds-hbp84 9h	1/1	Running	0
telegraf-ds-plwzv 9h	1/1	Running	0
telegraf-ds-sr22c 9h	1/1	Running	0
telegraf-rs-4sbg8 9h	1/1	Running	0
telemetry-service-fb9559f7b-mk917 (9h ago) 9h	1/1	Running	3
tenancy-559bbc6b48-5msgg 9h	1/1	Running	0
traefik-d997b8877-7xpf4 9h	1/1	Running	0
traefik-d997b8877-9xv96 9h	1/1	Running	0
trident-svc-585c97548c-d25z5 9h	1/1	Running	0
vault-controller-88484b454-2d6sr 9h	1/1	Running	0
vault-controller-88484b454-fc5cz 9h	1/1	Running	0
vault-controller-88484b454-jktld 9h	1/1	Running	0

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

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



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

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

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

Antwort:

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	23.04.2-7	10.111.111.111
True			



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

## Eindringen für den Lastenausgleich einrichten

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

Nachdem Astra Control Center bereitgestellt wurde, müssen Sie den Ingress-Controller so konfigurieren, dass Astra Control Center mit einer URL verfügbar ist.

Die Einstellungsschritte unterscheiden sich je nach Typ des Ingress-Controllers. Astra Control Center unterstützt viele Ingress-Controller-Typen. Diese Einstellungsverfahren enthalten Beispielschritte für die folgenden Ingress-Controller-Typen:

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

## Bevor Sie beginnen

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

## Schritte für Istio Ingress

1. Konfigurieren Sie Istio Ingress.



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

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

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

Beispielbefehl:

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

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

Beispielbefehl:

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



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

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

```

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

```

##### 5. Übernehmen Sie die Änderungen:

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

##### 6. Überprüfen Sie den Status des Eingangs:

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

Antwort:

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

## 7. Astra Control Center-Installation abschließen.

### Schritte für Nginx Ingress Controller

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

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

3. Übernehmen Sie die Änderungen:

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



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

### Schritte für OpenShift-Eingangs-Controller

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

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

## Melden Sie sich in der UI des Astra Control Center an

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

### Schritte

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



Sie können nach der Anmeldung ein benutzerdefiniertes Zertifikat erstellen.

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



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

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



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

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

## Beheben Sie die Fehlerbehebung für die Installation

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

### Optionen

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

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

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

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

## Wie es weiter geht

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

## Konfigurieren Sie einen externen Zertifikaten-Manager

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

### Schritte

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

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

Beispielantwort:

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

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

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

Beispielantwort:

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

3. Erstellen eines Geheimnisses mit zuvor generierten Dateien:



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

Beispielantwort:

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

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

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

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

Beispielantwort:

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

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

```
kubectl get ClusterIssuer
```

Beispielantwort:

NAME	READY	AGE
astra-ca-clusterissuer	True	9s

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

## Installieren Sie Astra Control Center mit OpenShift OperatorHub

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

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

### Bevor Sie beginnen

- **Voraussetzungen für die Umwelt erfüllt:** "[Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor](#)".

- **Gesunde Cluster-Betreiber 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
```

- **FQDN-Adresse:** Erhalten Sie eine FQDN-Adresse für Astra Control Center in Ihrem Rechenzentrum.
- **OpenShift Permissions:** Erhalten Sie die erforderlichen Berechtigungen und den Zugriff auf die Red hat OpenShift Container Platform, um die beschriebenen Installationsschritte durchzuführen.
- **Cert Manager konfiguriert:** Wenn bereits ein Cert Manager 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:** Wenn Sie über einen Kubernetes Ingress-Controller verfügen, der externen Zugriff auf Services wie etwa den Lastausgleich in einem Cluster managt, müssen Sie ihn zur Verwendung mit Astra Control Center einrichten:
  - a. Erstellen Sie den Operator-Namespace:

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

- b. "[Einrichtung abschließen](#)" Für Ihren Ingress-Controller-Typ.

### Schritte

- [Laden Sie das Astra Control Center herunter und extrahieren Sie es](#)
- [Installieren Sie das NetApp Astra kubectl Plug-in](#)
- [Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu](#)

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

### Laden Sie das Astra Control Center herunter und extrahieren Sie es

1. Wechseln Sie zum "[Download-Seite für Astra Control Center](#)" Auf der NetApp Support Site
2. Laden Sie das Bundle mit Astra Control Center herunter (`astra-control-center-[version].tar.gz`).
3. (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 Pakets 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.

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

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

### Installieren Sie das NetApp Astra kubectl Plug-in

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

#### Bevor Sie beginnen

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

#### Schritte

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



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

```
ls kubectl-astra/
```

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

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

### **Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu**

1. Führen Sie die entsprechende Schrittfolge für Ihre Container-Engine durch:

## Docker

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

```
acc.manifest.bundle.yaml
acc/
```

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

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

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

## Podman

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

```
acc.manifest.bundle.yaml
acc/
```

2. Melden Sie sich bei Ihrer Registrierung an:

```
podman login <YOUR_REGISTRY>
```

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

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

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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

```

**Podman 3**

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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:

```

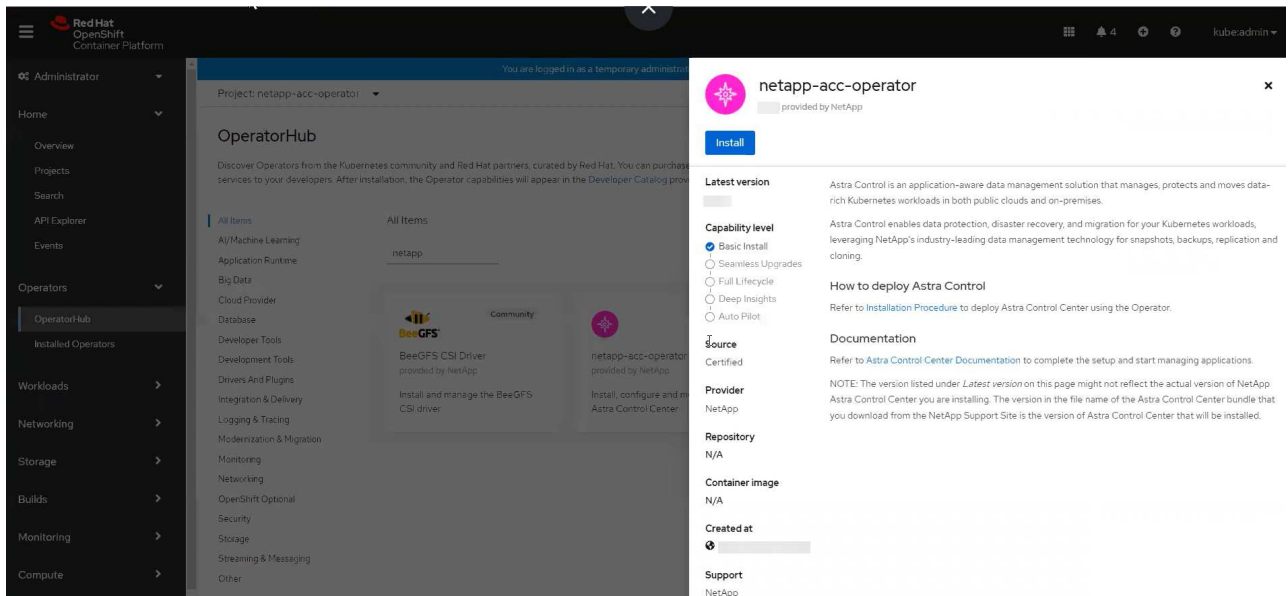
https://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version

```

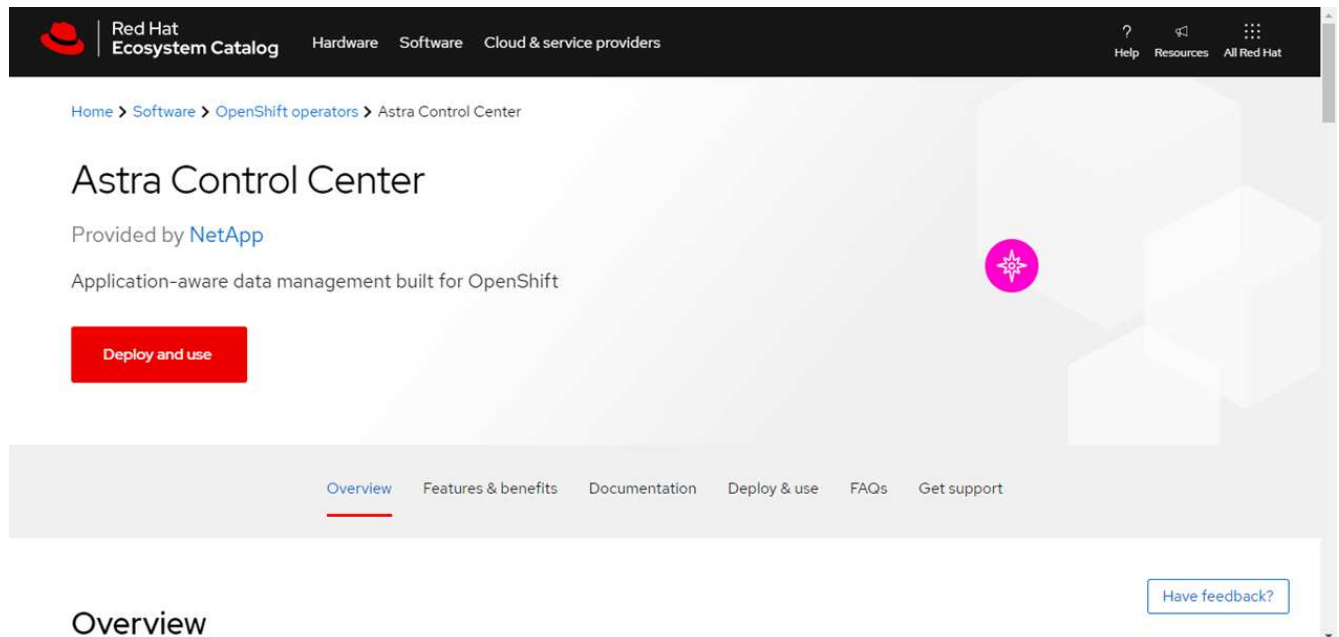
## Suchen Sie die Installationsseite des Bedieners

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

- Von der Red Hat OpenShift-Webkonsole aus:
  - i. Melden Sie sich in der OpenShift Container Platform UI an.
  - ii. Wählen Sie im Seitenmenü die Option **Operatoren > OperatorHub** aus.
  - iii. Suchen Sie nach und wählen Sie den Operator des NetApp Astra Control Center aus.



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



## Installieren Sie den Operator

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



Der Operator ist in allen Cluster-Namespace verfügbar.

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



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

- c. Wählen Sie **Installieren**.



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

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

## Installieren Sie Astra Control Center

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

The screenshot shows the 'Operator details' page for 'netapp-acc-operator' (version 23.4.0) in the 'netapp-acc-operator' project. The 'Astra Control Center' tab is active. Below the tabs, there is a section for 'AstraControlCenters' with a 'Show operands in:' dropdown set to 'All namespaces'. A blue button labeled 'Create AstraControlCenter' is present. The main content area displays 'No operands found' and a note: 'Operands are declarative components used to define the behavior of the application.'

2. Füllen Sie die aus `Create AstraControlCenter` Formularfeld:
  - a. Behalten Sie den Namen des Astra Control Center bei oder passen Sie diesen an.
  - b. Fügen Sie Etiketten für das Astra Control Center hinzu.
  - c. Aktivieren oder deaktivieren Sie Auto Support. Es wird empfohlen, die Auto Support-Funktion beizubehalten.
  - d. Geben Sie den FQDN des Astra Control Centers oder die IP-Adresse ein. Kommen Sie nicht herein `http://` Oder `https://` Im Adressfeld.
  - e. Geben Sie die Astra Control Center-Version ein, z. B. 23.04.2-7.
  - f. Geben Sie einen Kontonamen, eine E-Mail-Adresse und einen Administratorknamen ein.
  - g. Wählen Sie eine Richtlinie zur Rückgewinnung von Volumes aus `Retain`, `Recycle`, Oder `Delete`. Der Standardwert ist `Retain`.
  - h. Wählen Sie die `ScaleSize` der Installation aus.





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

i. Wählen Sie den Eingangstyp aus:

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

Verwenden Sie diese Option, wenn Sie einen anderen Ingress-Controller verwenden oder Ihren eigenen Ingress-Controller verwenden möchten. Nach der Implementierung des Astra Control Center müssen Sie den konfigurieren "**Eingangs-Controller**" Um Astra Control Center mit einer URL zu zeigen.

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

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

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



Weitere Informationen zum Servicetyp „loadbalancer“ und „ingress“ finden Sie unter "[Anforderungen](#)".

- a. Geben Sie in **Image Registry** Ihren lokalen Container Image Registry-Pfad ein. Kommen Sie nicht herein `http://` Oder `https://` Im Adressfeld.
- b. Wenn Sie eine Bildregistrierung verwenden, die eine Authentifizierung erfordert, geben Sie das Bildgeheimnis ein.



Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, [Erstellen Sie ein Geheimnis auf dem Cluster](#).

- c. Geben Sie den Vornamen des Administrators ein.
- d. Konfiguration der Ressourcenskalisierung
- 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, für die eine Authentifizierung erforderlich ist, erstellen Sie im OpenShift-Cluster ein Geheimnis, und geben Sie den geheimen Namen in ein `Create AstraControlCenter` Formularfeld.

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

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

2. Erstellen eines Geheimnisses in diesem Namespace:

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



Astra Control unterstützt nur die Geheimnisse der Docker-Registrierung.

3. Füllen Sie die übrigen Felder in aus [Das Feld AstraControlCenter-Formular erstellen](#).

### Wie es weiter geht

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

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

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

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

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

- [Implementieren Sie Astra Control Center in Amazon Web Services](#)
- [Implementieren Sie Astra Control Center in der Google Cloud Platform](#)
- [Implementieren Sie Astra Control Center in Microsoft Azure](#)

Applikationen lassen sich in Distributionen mit selbst gemanagten Kubernetes-Clustern managen, wie z. B. mit

OpenShift Container Platform (OCP). Nur selbst gemanagte OCP Cluster sind für die Implementierung des Astra Control Center validiert.

## Implementieren Sie Astra Control Center in Amazon Web Services

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

### Was Sie für AWS benötigen

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

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

### Anforderungen der Betriebsumgebung für AWS

Astra Control Center erfordert die folgende Betriebsumgebung für AWS:


- Red hat OpenShift Container Platform 4.8



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

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

Komponente	Anforderungen
<b>Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität</b>	Mindestens 300 GB verfügbar
<b>Worker-Nodes (AWS EC2 Anforderung)</b>	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
<b>Load Balancer</b>	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
<b>FQDN</b>	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse

Komponente	Anforderungen
<b>Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP, ehemals Cloud Manager)</b>	Astra Trident 21.04 oder höher ist installiert und konfiguriert und NetApp ONTAP Version 9.5 oder höher als Storage-Backend
<b>Bildregistrierung</b>	<p>Sie müssen über eine vorhandene private Registry, wie AWS Elastic Container Registry, mit der Sie Astra Control Center Build-Images übertragen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  <p>Der gehostete Astra Control Center-Cluster und der verwaltete Cluster müssen Zugriff auf dieselbe Image-Registry haben, um Anwendungen mit dem Restic-basierten Image sichern und wiederherstellen zu können.</p> </div>
<b>Konfiguration von Astra Trident/ONTAP</b>	<p>Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden Kubernetes-Storage-Klassen von ONTAP, die beim Importieren des Kubernetes Clusters in NetApp BlueXP (ehemals Cloud Manager) erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-san</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-san</code> <code>csi.trident.netapp.io</code></li> </ul>



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



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

### Überblick über die Implementierung für AWS

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

Jeder dieser Schritte wird unten im Detail erklärt.

1. [dass Sie über ausreichende IAM-Berechtigungen verfügen.](#)

2. [Installation eines RedHat OpenShift-Clusters in AWS.](#)
3. [Konfigurieren von AWS.](#)
4. [Konfiguration von NetApp BlueXP für AWS.](#)
5. [Installieren Sie Astra Control Center für AWS.](#)

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

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

Siehe "[Erste AWS Zugangsdaten](#)".

#### **Installation eines RedHat OpenShift-Clusters in AWS**

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

Installationsanweisungen finden Sie unter "[Installation eines Clusters auf AWS in OpenShift Container Platform](#)".

#### **Konfigurieren von AWS**

Konfigurieren Sie dann AWS für die Erstellung eines virtuellen Netzwerks, richten Sie EC2 Computing-Instanzen ein, erstellen Sie einen AWS S3-Bucket, erstellen Sie ein Elastic Container Register (ECR), um die Astra Control Center Images zu hosten und übertragen Sie die Images auf diese Registrierung.

Folgen Sie der AWS Dokumentation, um die folgenden Schritte auszuführen. Siehe "[AWS Installationsdokumentation](#)".

1. Virtuelles AWS Netzwerk erstellen.
2. EC2 Computing-Instanzen prüfen. Dabei können es sich um einen Bare Metal Server oder VMs in AWS handeln.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in AWS, um die Astra-Anforderungen zu erfüllen. Siehe "[Anforderungen des Astra Control Centers](#)".
4. Erstellen Sie mindestens einen AWS S3-Bucket zum Speichern Ihrer Backups.
5. AWS Elastic Container Registry (ECR) erstellen, um alle ACC-Images zu hosten



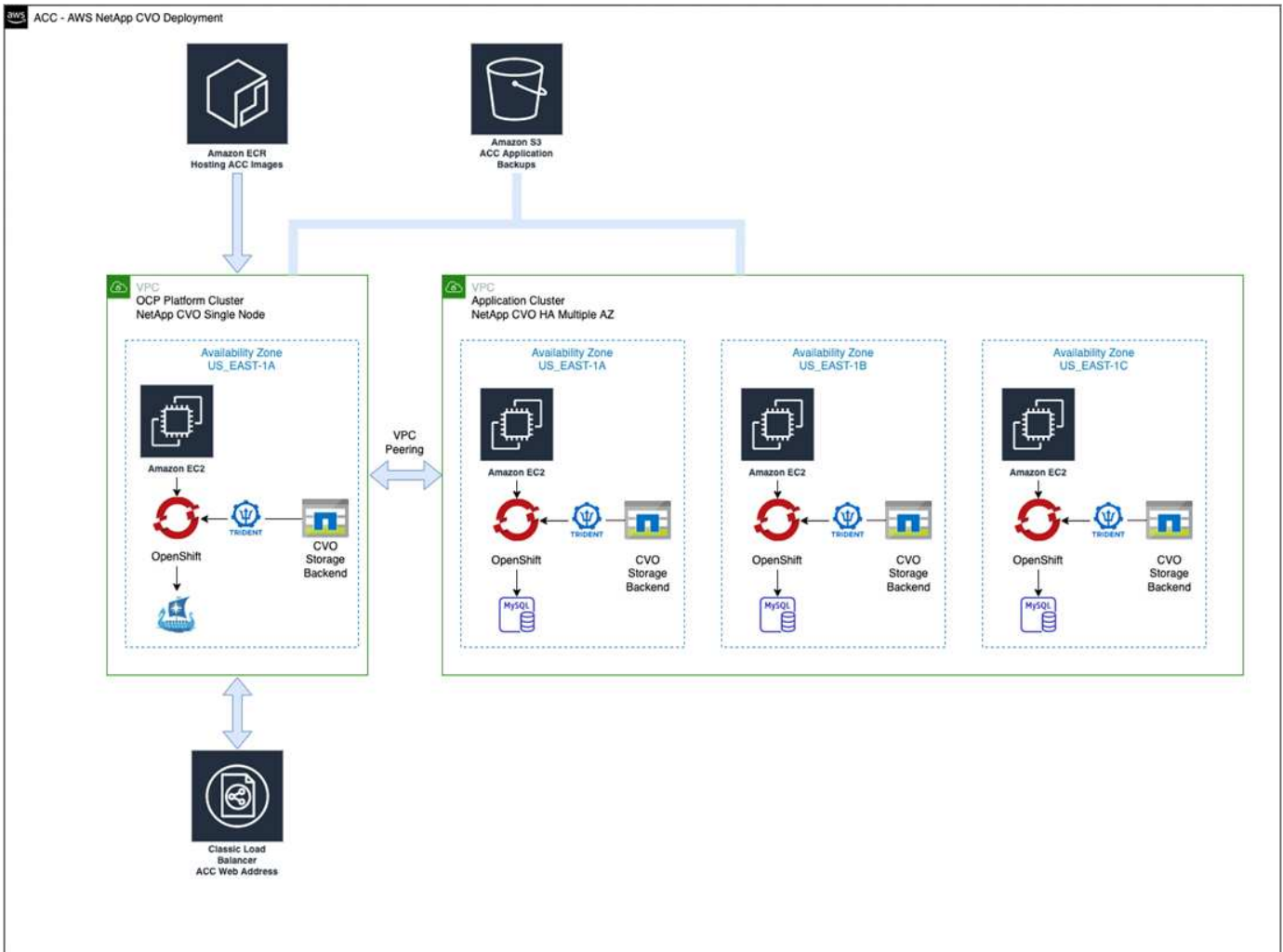
Wenn Sie den ECR nicht erstellen, kann Astra Control Center mit einem AWS Backend nicht auf die Monitoring-Daten von einem Cluster mit Cloud Volumes ONTAP zugreifen. Das Problem wird verursacht, wenn der Cluster, den Sie mit Astra Control Center ermitteln und verwalten möchten, keinen AWS ECR-Zugriff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

6. Drücken Sie die ACC-Bilder auf die definierte Registrierung.



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

Beispiel für eine AWS Implementierung:



## Konfiguration von NetApp BlueXP für AWS

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

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

- ["Erste Schritte mit Cloud Volumes ONTAP in AWS"](#).
- ["Erstellen Sie einen Connector in AWS mit BlueXP"](#)

## Schritte

1. Fügen Sie Ihre Anmeldeinformationen zu BlueXP hinzu.
2. Erstellen Sie einen Arbeitsbereich.
3. Fügen Sie einen Connector für AWS hinzu. Entscheiden Sie sich für AWS als Provider.
4. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
  - a. Ort: „Amazon Web Services (AWS)“
  - b. Typ: „Cloud Volumes ONTAP HA“
5. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.

- a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s > Cluster list > Cluster-Details** wählen.
- b. Beachten Sie in der oberen rechten Ecke die Astra Trident-Version.
- c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen, für die NetApp als provisionierung angezeigt wird.

Dies importiert Ihr Red hat OpenShift-Cluster und weist ihm eine Standardspeicherklasse zu. Sie wählen die Speicherklasse aus.

Astra Trident wird automatisch im Rahmen des Imports und der Erkennung installiert.

6. Beachten Sie alle persistenten Volumes und Volumes in dieser Cloud Volumes ONTAP-Implementierung.



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

### Installieren Sie Astra Control Center für AWS

Dem Standard folgen "[Installationsanweisungen für Astra Control Center](#)".



AWS verwendet den Bucket-Typ generischer S3.

### Implementieren Sie Astra Control Center in der Google Cloud Platform

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

#### Was wird für GCP benötigt

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


- Astra Control Center-Lizenz: Siehe "[Lizenzierungsanforderungen für Astra Control Center](#)".
- "[Sie erfüllen die Anforderungen des Astra Control Centers](#)".
- NetApp Cloud Central Konto
- Bei Verwendung von OCP, Red hat OpenShift Container Platform (OCP) 4.10
- Bei Verwendung von OCP, Berechtigungen für die Red hat OpenShift Container Platform (OCP) (auf Namespace-Ebene zum Erstellen von Pods)
- GCP-Servicekonto mit Berechtigungen, mit denen Sie Buckets und Konnektoren erstellen können

#### Anforderungen an die Betriebsumgebung für GCP



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

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

Komponente	Anforderungen
<b>Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität</b>	Mindestens 300 GB verfügbar
<b>Worker-Nodes (GCP-Compute-Anforderung)</b>	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
<b>Load Balancer</b>	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
<b>FQDN (GCP-DNS-ZONE)</b>	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
<b>Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP, ehemals Cloud Manager)</b>	Astra Trident 21.04 oder höher ist installiert und konfiguriert und NetApp ONTAP Version 9.5 oder höher als Storage-Backend
<b>Bildregistrierung</b>	<p>Sie müssen über eine bestehende private Registrierung, wie Google Container Registry, zu denen Sie Astra Control Center Bilder erstellen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Sie müssen anonymen Zugriff aktivieren, um Restic Images für Backups zu erstellen. </div>
<b>Konfiguration von Astra Trident/ONTAP</b>	<p>Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden ONTAP Kubernetes Storage-Klassen, die beim Import des Kubernetes Clusters in NetApp BlueXP erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-san</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-san</code> <code>csi.trident.netapp.io</code></li> </ul>



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



## Übersicht über die Implementierung für GCP

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

Jeder dieser Schritte wird unten im Detail erklärt.

1. [Installieren Sie einen RedHat OpenShift-Cluster auf GCP.](#)
2. [Erstellung eines GCP-Projekts und einer virtuellen Private Cloud.](#)
3. [dass Sie über ausreichende IAM-Berechtigungen verfügen.](#)
4. [GCP konfigurieren.](#)
5. [NetApp BlueXP für GCP konfigurieren.](#)
6. [Astra Control Center für GCP installieren.](#)

### Installieren Sie einen RedHat OpenShift-Cluster auf GCP

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

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

- ["Installation eines OpenShift-Clusters in GCP"](#)
- ["Erstellen eines GCP-Service-Kontos"](#)

### Erstellung eines GCP-Projekts und einer virtuellen Private Cloud

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



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

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

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

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

Siehe ["Erste GCP-Zugangsdaten und -Berechtigungen"](#).

### GCP konfigurieren

Konfigurieren Sie dann GCP zur Erstellung einer VPC, zur Einrichtung von Computing-Instanzen, zur Erstellung eines Google Cloud Objekt-Storage, zur Erstellung eines Google-Container-Registers für das Hosten der Astra Control Center-Images und zum Senden der Bilder an diese Registry.

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

Beispiel: ACC-Bilder können durch Eingabe des folgenden Skripts in diese Registrierung verschoben werden:

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

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 < astra-control-center-22.04.41.manifest
```

8. Richten Sie DNS-Zonen ein.

### NetApp BlueXP für GCP konfigurieren

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

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

### Bevor Sie beginnen

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

### Schritte

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

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



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

### Astra Control Center für GCP installieren

Dem Standard folgen ["Installationsanweisungen für Astra Control Center"](#).



GCP verwendet den allgemeinen S3-Bucket-Typ.

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

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

### Implementieren Sie Astra Control Center in Microsoft Azure

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

#### Was Sie für Azure benötigen

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

- Astra Control Center-Lizenz: Siehe ["Lizenzierungsanforderungen für Astra Control Center"](#).

- ["Sie erfüllen die Anforderungen des Astra Control Centers"](#).
- NetApp Cloud Central Konto
- Bei Verwendung von OCP, Red hat OpenShift Container Platform (OCP) 4.8
- Bei Verwendung von OCP, Berechtigungen für die Red hat OpenShift Container Platform (OCP) (auf Namespace-Ebene zum Erstellen von Pods)
- Azure Zugangsdaten mit Berechtigungen, mit denen Sie Buckets und Konnektoren erstellen können

#### Anforderungen an die Betriebsumgebung für Azure

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

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

Siehe ["Anforderungen an die Betriebsumgebung des Astra Control Centers"](#).

Komponente	Anforderungen
<b>Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität</b>	Mindestens 300 GB verfügbar
<b>Worker-Nodes (Azure-Computing-Anforderung)</b>	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
<b>Load Balancer</b>	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
<b>FQDN (Azure-DNS-Zone)</b>	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
<b>Astra Trident (installiert im Rahmen der Kubernetes Cluster Discovery in NetApp BlueXP)</b>	Astra Trident 21.04 oder neuer installiert und konfiguriert und NetApp ONTAP Version 9.5 oder neuer wird als Storage-Backend verwendet
<b>Bildregistrierung</b>	<p>Sie müssen über eine vorhandene private Registry, wie z. B. Azure Container Registry (ACR) verfügen, in die Sie Bilder vom Astra Control Center erstellen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen.</p> <div style="display: flex; align-items: center; margin-top: 10px;">  <p>Sie müssen anonymen Zugriff aktivieren, um Restic Images für Backups zu erstellen.</p> </div>

Komponente	Anforderungen
<b>Konfiguration von Astra Trident/ONTAP</b>	<p>Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden ONTAP Kubernetes Storage-Klassen, die beim Import des Kubernetes Clusters in NetApp BlueXP erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-ha-san</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-nas</code> <code>csi.trident.netapp.io</code></li> <li>• <code>vsaworkingenvironment-&lt;&gt;-single-san</code> <code>csi.trident.netapp.io</code></li> </ul>



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

### Überblick über die Implementierung für Azure

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

Jeder dieser Schritte wird unten im Detail erklärt.

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

### Installieren Sie einen RedHat OpenShift-Cluster auf Azure

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

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

- ["OpenShift-Cluster wird auf Azure installiert".](#)
- ["Installieren eines Azure-Kontos".](#)

### Erstellen von Azure Ressourcengruppen

Erstellen Sie mindestens eine Azure-Ressourcengruppe.



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

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

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

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

Siehe "[Azure Zugangsdaten und Berechtigungen](#)".

#### **Konfigurieren Sie Azure**

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

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

1. Virtuelles Azure Netzwerk erstellen.
2. Prüfen Sie die Computing-Instanzen. Dabei können es sich um einen Bare Metal Server oder VMs in Azure handeln.
3. Wenn der Instanztyp nicht bereits den Mindestanforderungen für Ressourcen von Astra für Master- und Worker-Nodes entspricht, ändern Sie den Instanztyp in Azure, um die Astra-Anforderungen zu erfüllen. Siehe "[Anforderungen des Astra Control Centers](#)".
4. Erstellen Sie mindestens einen Azure Blob Container, um Ihre Backups zu speichern.
5. Erstellen Sie ein Speicherkonto. Sie benötigen ein Storage-Konto, um einen Container zu erstellen, der im Astra Control Center als Bucket verwendet wird.
6. Erstellen eines Geheimnisses, das für den Bucket-Zugriff erforderlich ist
7. Erstellen Sie eine Azure Container Registry (ACR), um alle Astra Control Center-Images zu hosten.
8. ACR-Zugriff für Docker-Push/Pull-alle Astra Control Center-Images einrichten.
9. Drücken Sie die ACC-Bilder in diese Registrierung, indem Sie das folgende Skript eingeben:

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

- Beispiel\*:

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

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

10. 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

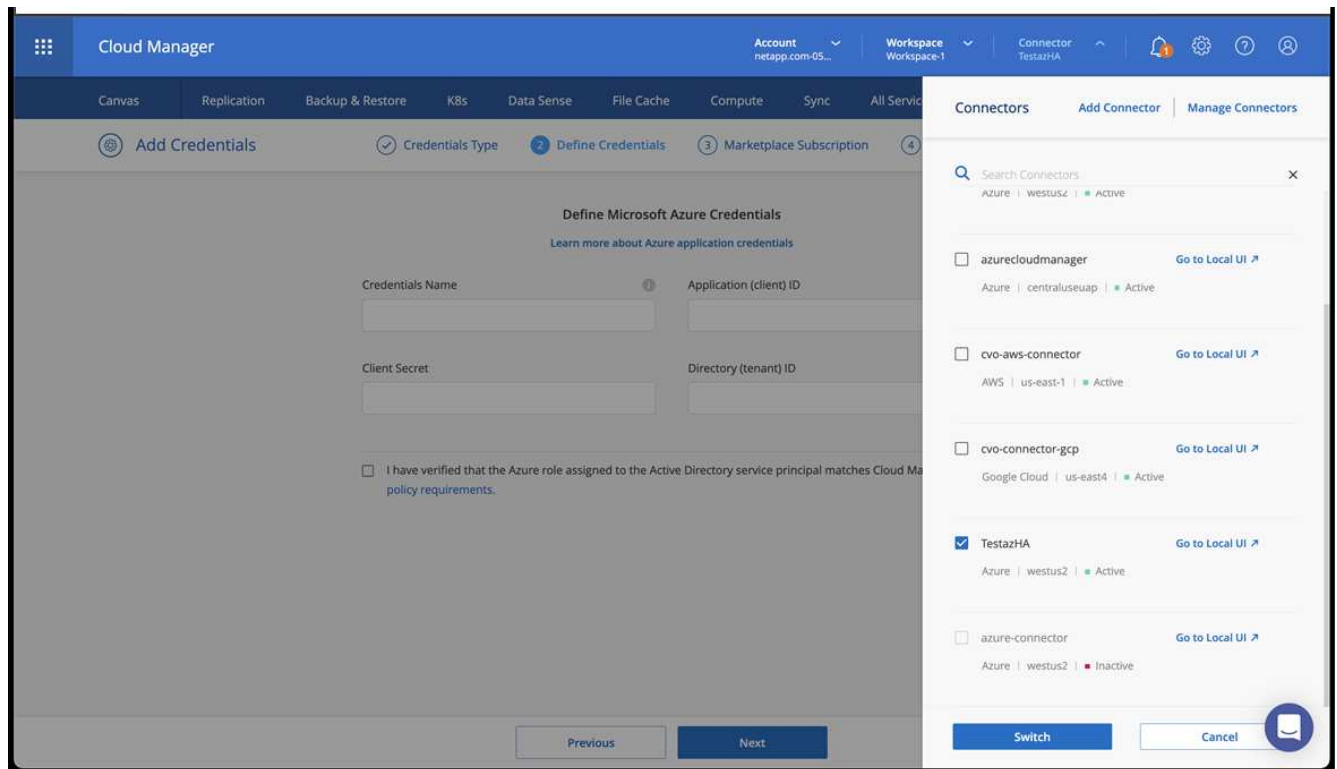
Zugriff auf das Azure Konto mit den erforderlichen IAM-Berechtigungen und -Rollen

#### 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 BlueXP](#)".

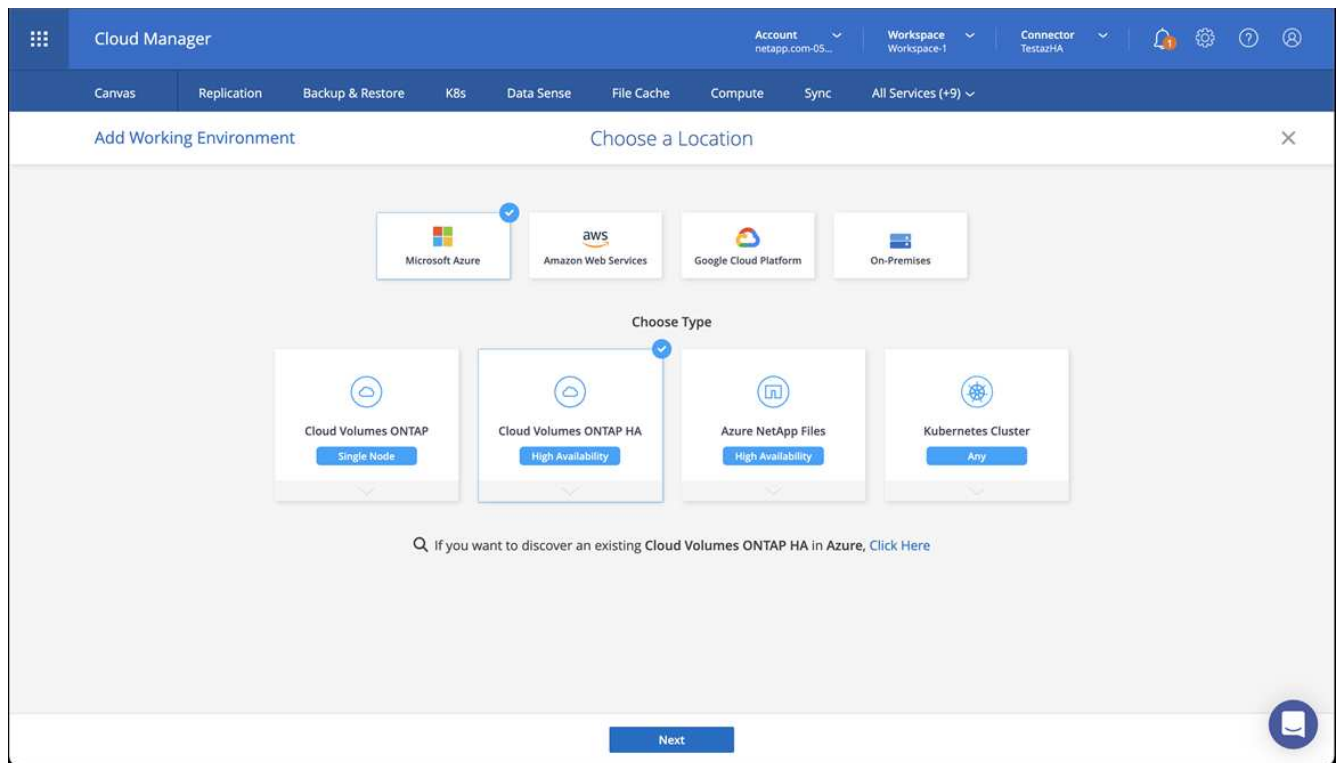
3. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.



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

a. Ort: „Microsoft Azure“.

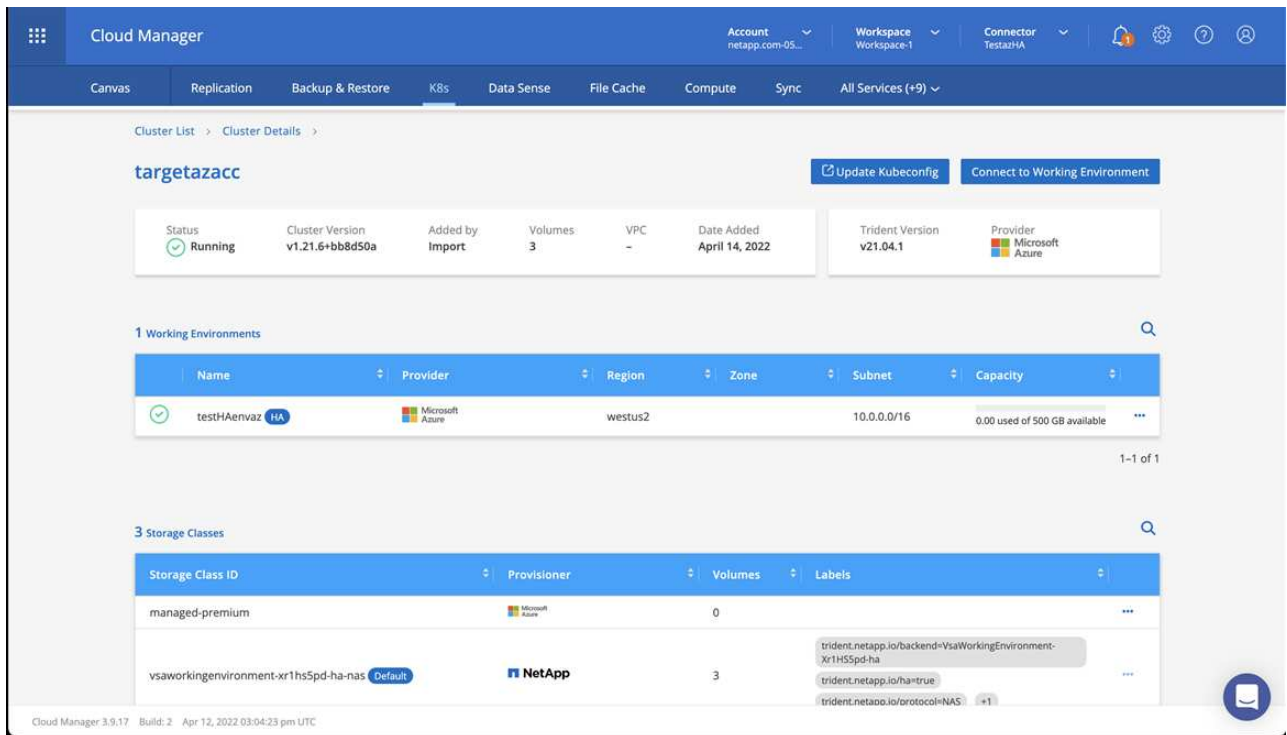
b. Typ: „Cloud Volumes ONTAP HA“.



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



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



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

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

Damit wird Ihr Red hat OpenShift-Cluster importiert und eine Standardspeicherklasse zugewiesen. Sie wählen die Speicherklasse aus.

Astra Trident wird automatisch im Rahmen des Imports und der Erkennung installiert.

6. Beachten Sie alle persistenten Volumes und Volumes in dieser Cloud Volumes ONTAP-Implementierung.

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

### Installation und Konfiguration von Astra Control Center für Azure

Installieren Sie Astra Control Center standardmäßig ["Installationsanweisungen"](#).

Fügen Sie über Astra Control Center einen Azure-Bucket hinzu. Siehe ["Astra Control Center einrichten und Buckets hinzufügen"](#).

### Konfigurieren Sie nach der Installation das Astra Control Center

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

### Ressourceneinschränkungen entfernen

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

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

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

### Schritte

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

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

Antwort:

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

2. Alle Ressourcen-Kontingente nach Namen löschen:

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

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

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

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

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

Antwort:

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

#### 4. Grenzwerte nach Namen löschen:

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

### Aktivieren der Netzwerkkommunikation zwischen Namespaces

Einige Umgebungen verwenden NetworkPolicy-Konstrukte, um den Datenverkehr zwischen Namespaces zu beschränken. Der Astra Control Center Operator und das Astra Control Center sind in verschiedenen Namespaces. Die Dienste in diesen verschiedenen Namespaces müssen in der Lage sein, miteinander zu kommunizieren. Gehen Sie wie folgt vor, um diese Kommunikation zu aktivieren.

#### Schritte

1. Löschen Sie alle im Astra Control Center Namespace vorhandenen NetworkPolicy-Ressourcen:

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

2. Verwenden Sie für jedes NetworkPolicy-Objekt, das vom vorhergehenden Befehl zurückgegeben wird, den folgenden Befehl, um es zu löschen. Ersetzen Sie [OBJECT\_NAME] durch den Namen des zurückgegebenen Objekts:

```
kubectl delete networkpolicy [OBJECT_NAME] -n [netapp-acc or custom namespace]
```

3. Wenden Sie die folgende Ressourcendatei an, um den zu konfigurieren `acc-avp-network-policy`. Soll Astra Plug-in-Services Anfragen an Astra Control Center Services stellen können. Ersetzen Sie die Informationen in Klammern <> durch Informationen aus Ihrer Umgebung:

```

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-avp-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            kubernetes.io/metadata.name: <PLUGIN_NAMESPACE_NAME> #
REPLACE THIS WITH THE ASTRA PLUGIN NAMESPACE NAME

```

4. Wenden Sie die folgende Ressourcendatei an, um den zu konfigurieren acc-operator-network-policy Objekt, um dem Astra Control Center-Betreiber die Kommunikation mit den Astra Control Center-Diensten zu ermöglichen. Ersetzen Sie die Informationen in Klammern <> durch Informationen aus Ihrer Umgebung:

```

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-operator-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            kubernetes.io/metadata.name: <NETAPP-ACC-OPERATOR> #
REPLACE THIS WITH THE OPERATOR NAMESPACE NAME

```

### Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

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

Das selbstsignierte Standardzertifikat wird für zwei Verbindungstypen verwendet:



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

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

### Bevor Sie beginnen

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

### Entfernen Sie das selbstsignierte Zertifikat

Entfernen Sie das vorhandene selbstsignierte TLS-Zertifikat.

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

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

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

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

### Fügen Sie mithilfe der Befehlszeile ein neues Zertifikat hinzu

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

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

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

2. Verwenden Sie den folgenden Befehl und das folgende Beispiel, um die Cluster-Datei CRD (Custom Resource Definition) zu bearbeiten und die zu ändern `spec.selfSigned` Mehrwert für

`spec.ca.secretName` So verweisen Sie auf das zuvor erstellte TLS-Geheimnis:

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

#spec:
#  selfSigned: {}

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

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

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

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

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

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

5. Erstellen Sie das Zertifikat mit dem folgenden Befehl:

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

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

```

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

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

```

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



```

kubectrl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

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

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

```

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

## Einrichten des Astra Control Center

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.

### Aufgaben

- [Fügen Sie eine Lizenz für Astra Control Center hinzu](#)
- [Bereiten Sie Ihre Umgebung mit Astra Control auf das Cluster-Management vor](#)
- [Cluster hinzufügen](#)
- [Aktivieren Sie die Authentifizierung auf dem ONTAP Storage Back-End](#)
- [Fügen Sie ein Storage-Back-End hinzu](#)
- [Fügen Sie einen Bucket hinzu](#)

### 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 "[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.

## 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 Rollen für das Cluster-Management schafft.

### Bevor Sie beginnen

- Stellen Sie sicher, dass die Worker-Nodes in Ihrem Cluster mit den entsprechenden Storage-Treibern konfiguriert sind, damit die Pods mit dem Back-End Storage interagieren können.
- Ihre Umgebung erfüllt die Anforderungen "[Anforderungen an die Betriebsumgebung](#)" Für Astra Trident und Astra Control Center.
- Eine Version von Astra Trident ist das "[Unterstützt durch Astra Control Center](#)" installiert:



Das können Sie "[Implementieren Sie Astra Trident](#)" mit dem Astra Trident Operator (manuell oder mit dem Helm Chart) oder `tridentctl`. Vor der Installation oder dem Upgrade von Astra Trident sollten Sie sich die "[Unterstützte Frontends, Back-Ends und Host-Konfigurationen](#)".

- **Astra Trident Storage-Backend konfiguriert:** Es muss mindestens ein Astra Trident Storage-Backend sein "Konfiguriert" Auf dem Cluster.
  - **Konfigurierte Astra Trident Storage-Klassen:** Es muss mindestens eine Astra Trident Storage-Klasse sein "Konfiguriert" Auf dem Cluster. Wenn eine Standard-Storage-Klasse konfiguriert ist, stellen Sie sicher, dass diese die einzige Storage-Klasse mit der Standardbeschriftung ist.
  - **Astra Trident Volume Snapshot Controller und Volume Snapshot Klasse installiert und konfiguriert:** Der Volume Snapshot Controller muss sein "Installiert" Damit Snapshots in Astra Control erstellt werden können. Mindestens ein Astra Trident `VolumeSnapshotClass` Gewesen "Einrichtung" Durch einen Administrator.
- **Kubeconfig:** Sie haben Zugang zum "Cluster kubeconfig" Das umfasst nur ein Kontextseil.
  - **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
```

- **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 Stuebenen-Kontext anstelle des Rancher API-Serverkontexts zu verwenden. So wird die Last auf dem Rancher API Server reduziert und die Performance verbessert.

## Führen Sie Eignungsprüfungen durch

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

### Schritte

1. Testen Sie die Version von Astra Trident.

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

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

NAME	VERSION
trident	22.10.0

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

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



Wenn Astra Trident nicht installiert ist oder die installierte Version nicht die neueste ist, müssen Sie die neueste Version von Astra Trident installieren, bevor Sie fortfahren. Siehe "[Astra Trident-Dokumentation](#)" Weitere Anweisungen.

2. Stellen Sie sicher, dass die Pods ausgeführt werden:

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

3. 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
true	5d23h	Immediate

## Erstellen Sie eine begrenzte Cluster-Rolle kubeconfig

Optional können Sie eine eingeschränkte Administratorrolle für Astra Control Center erstellen. Dies ist kein erforderliches Verfahren für die Einrichtung des Astra Control Centers. Dieses Verfahren hilft beim Erstellen eines separaten kubeconfig, das die Astra Control-Berechtigungen auf die von ihm verwalteten Cluster beschränkt.

### 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`.

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

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

+

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

- a. Wenden Sie das Servicekonto an:

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

### 2. Erstellen Sie eine begrenzte Cluster-Rolle mit den minimalen Berechtigungen, die für das Management eines Clusters durch Astra Control erforderlich sind:

- a. Erstellen Sie ein `ClusterRole` Datei aufgerufen `astra-admin-account.yaml`.

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

```
<strong>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
  - podsecuritypolicies
  - replicaset
  - 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

# Use PodSecurityPolicies
- apiGroups:
  - extensions
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use

```

a. 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.

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

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

+

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

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

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

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

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

Das Ende der Ausgabe sollte wie folgt aussehen:



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

Die Indizes für jedes Element im `secrets` Array beginnt mit 0. Im obigen Beispiel der Index für `astracontrol-service-account-dockercfg-vhz87` wäre 0 und der Index für `astracontrol-service-account-token-r59kr` sind es 1. Notieren Sie in Ihrer Ausgabe den Index für den Namen des Dienstkontos, der das Wort „Token“ darin enthält.

5. Erzeugen Sie den `kubeconfig` wie folgt:

- a. Erstellen Sie ein `create-kubeconfig.sh` Datei: 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

```

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

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

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

## Was kommt als Nächstes?

Nachdem Sie nun überprüft haben, ob die Voraussetzungen erfüllt sind, können Sie es jetzt tun [Fügen Sie einen Cluster hinzu](#).

## Cluster hinzufügen

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



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

### Bevor Sie beginnen

- Bevor Sie ein Cluster hinzufügen, überprüfen und führen Sie die erforderlichen Maßnahmen durch [Erforderliche Aufgaben](#).

### 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 `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 [Das oben beschriebene Verfahren](#), 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.



Sie sollten eine Astra Trident Storage-Klasse auswählen, die von ONTAP Storage unterstützt wird.

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. Um das Problem zu beheben, müssen Sie sich die Protokolle des Überwachungspers ansehen.

## Aktivieren Sie die Authentifizierung auf dem 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.

Details in `Beispiel.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 vservers 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:

Details in Beispiel.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
- **Vertrauenswürdige CA-Zertifikat:** ontap\_Signed\_cert.crt

## Fügen Sie ein Storage-Back-End hinzu


Sie können zum Managen Ihrer Ressourcen ein vorhandenes ONTAP-Storage-Backend zum Astra Control Center hinzufügen.

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.

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.

## 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 "[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

- Ein Bucket, der von Ihren Clustern erreichbar ist, die von Astra Control Center verwaltet werden.
- Zugangsdaten für den Bucket.
- Ein Bucket der folgenden Typen:
  - 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](#)".

## Was kommt als Nächstes?

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

- "Managen Sie lokale Benutzer und Rollen"
- "Starten Sie das Anwendungsmanagement"
- "Schützen von Applikationen"
- "Benachrichtigungen verwalten"
- "Verbinden Sie sich mit Cloud Insights"
- "Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu"
- "Ändern der Standard-Storage-Klasse"

## Weitere Informationen

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

## Häufig gestellte Fragen zum Astra Control Center

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

### Überblick

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

### Zugang zum Astra Control Center

#### Was ist die Astra Control URL?

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

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

### Lizenzierung

#### Ich verwende eine Evaluierungslizenz. Wie ändere ich die Volllizenz?

Sie können ganz einfach zu einer vollständigen Lizenz wechseln, indem Sie die NetApp Lizenzdatei (NetApp License File, NLF) von NetApp beziehen.

#### Schritte

1. Wählen Sie in der linken Navigationsleiste **Konto > Lizenz**.
2. Wählen Sie in der Lizenzübersicht rechts neben den Lizenzinformationen das Menü Optionen.
3. Wählen Sie **Ersetzen**.
4. Navigieren Sie zu der Lizenzdatei, die Sie heruntergeladen haben, und wählen Sie **Hinzufügen**.

## Ich verwende eine Evaluierungslizenz. Kann ich trotzdem Apps verwalten?

Ja, Sie können die Funktionalität zum Verwalten von Apps mit einer Evaluierungslizenz testen (einschließlich der standardmäßig installierten integrierten Evaluierungslizenz). Zwischen einer Evaluierungslizenz und einer vollständigen Lizenz gibt es keinen Unterschied in den Funktionen oder der Funktionalität; die Evaluierungslizenz hat einfach eine kürzere Lebensdauer. Siehe "[Lizenzierung](#)". Finden Sie weitere Informationen.

## Kubernetes Cluster werden registriert

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

Vorhandenen Pools können neue Worker Nodes hinzugefügt werden. Diese werden automatisch von Astra Control entdeckt. Wenn die neuen Knoten in Astra Control nicht sichtbar sind, prüfen Sie, ob auf den neuen Worker Nodes der unterstützte Bildtyp ausgeführt wird. Sie können den Zustand der neuen Worker-Nodes auch mit überprüfen `kubect1 get nodes` Befehl.

### Wie entnehme ich einen Cluster richtig?

1. "[Lösen Sie die Anwendungen von Astra Control](#)".
2. "[Lösen Sie das Cluster über Astra Control](#)".

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

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



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

### Wird NetApp Astra Trident automatisch aus einem Cluster deinstalliert, wenn ich es entmanage?

Wenn Sie ein Cluster aus Astra Control Center deinstallieren, wird Astra Trident nicht automatisch aus dem Cluster deinstalliert. Um Astra Trident zu deinstallieren, müssen Sie es benötigen "[Folgen Sie den Schritten in der Dokumentation von Astra Trident](#)".

## Management von Applikationen

### Kann Astra Control eine Anwendung bereitstellen?

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

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

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

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

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

### **Sollte ich Astra Control selbst managen?**

Nein, Astra Control sollte nicht selbst gemanagt werden, da es sich um eine „System-Anwendung“ handelt.

### **Wirken sich ungesunde Pods auf das App-Management aus?**

Nein, der Zustand von Pods beeinträchtigt das App-Management nicht.

## **Datenmanagement-Vorgänge**

### **Meine Anwendung verwendet mehrere PVS. Wird Astra Control Snapshots und Backups dieser PVS erstellen?**

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

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

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



## Copyright-Informationen

Copyright © 2023 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers 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, BEISPIELHAFT 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 IRGEND EINER 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.