



Los geht's

Astra Control Center

NetApp
June 06, 2024

Inhalt

- Los geht's 1
 - Anforderungen des Astra Control Centers 1
 - Schnellstart für Astra Control Center 7
 - Übersicht über die Installation 9
 - Einrichten des Astra Control Center 57
 - Häufig gestellte Fragen zum Astra Control Center 77

Los geht's

Anforderungen des Astra Control Centers

Prüfen Sie zunächst die Bereitschaft Ihrer Betriebsumgebung, Anwendungscluster, Applikationen, Lizenzen und Ihres Webbrowsers.

- [Anforderungen an die Betriebsumgebung](#)
- [Unterstützte Storage-Back-Ends](#)
- [Anforderungen für Applikationscluster](#)
- [Anforderungen für das Applikationsmanagement](#)
- [Replikationsvoraussetzungen](#)
- [Zugang zum Internet](#)
- [Lizenz](#)
- [Ingress für lokale Kubernetes Cluster](#)
- [Netzwerkanforderungen](#)
- [Unterstützte Webbrowser](#)

Anforderungen an die Betriebsumgebung

Astra Control Center wurde mit folgenden Typen von Betriebsumgebungen validiert:

- Google Anthos 1.10 oder 1.11
- Kubernetes 1.22 auf 1.24
- Rancher Kubernetes Engine (RKE):
 - RKE 1.2.16 mit Rancher 2.5.12 und RKE 1.3.3 mit 2.6.3
 - RKE 2 (v1.23,6+rke2r2) mit Rancher 2.6.3
- Red hat OpenShift Container Platform 4.8, 4.9 oder 4.10
- VMware Tanzu Kubernetes Grid 1.4 oder 1.5
- VMware Tanzu Kubernetes Grid Integrated Edition 1.12.2 oder 1.13

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

Komponente	Anforderungen
Storage-Back-End-Kapazität	Mindestens 500 GB verfügbar
Worker-Nodes	Insgesamt mindestens 3 Worker-Nodes mit 4 CPU-Kernen und jeweils 12 GB RAM
FQDN-Adresse	Eine FQDN-Adresse für Astra Control Center

Komponente	Anforderungen
Astra Trident	Astra Trident 21.10.1 oder höher ist installiert und konfiguriert Astra Trident 22.07 oder höher für SnapMirror-basierte Applikationsreplizierung



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.

- **Image Registry:** Sie benötigen eine bereits vorhandene private Docker-Image-Registry, mit der Sie Astra Control Center-Bilder erstellen können. Sie müssen die URL der Bildregistrierung angeben, in der Sie die Bilder hochladen.
- **Astra Trident / ONTAP Konfiguration:** Astra Control Center erfordert, dass eine Storage-Klasse erstellt und als Standard-Storage-Klasse eingestellt wird. Astra Control Center unterstützt die folgenden ONTAP-Treiber von Astra Trident:
 - ontap-nas
 - ontap-san
 - ontap-san-Ökonomie



Beim Klonen von Applikationen in OpenShift-Umgebungen muss das Astra Control Center OpenShift erlauben, Volumes anzuhängen und die Eigentümerschaft von Dateien zu ändern. Daher müssen Sie eine ONTAP Volume Export-Richtlinie konfigurieren, damit diese Vorgänge möglich sind. Sie können dies mit folgenden Befehlen tun:

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



Wenn Sie eine zweite OpenShift-Betriebsumgebung als gemanagte Computing-Ressource hinzufügen möchten, müssen Sie sicherstellen, dass die Astra Trident Volume Snapshot-Funktion aktiviert ist. Um Volume Snapshots mit Astra Trident zu aktivieren und zu testen, ["Sehen Sie sich die offiziellen Anweisungen von Astra Trident an"](#).

Cluster-Anforderungen für VMware Tanzu Kubernetes Grid

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.

- 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.
- Im Rahmen der Installation von Astra Control Center werden die folgenden Ressourcen in einer Pod Security Policy (PSP)-Umgebung mit eingeschränkter Sicherheit erstellt:
 - POD-Sicherheitsrichtlinie

- RBAC-Rolle
- RBAC Rolle und rollenbindende Ressourcen werden in der erstellt `netapp-acc` Namespace.
- 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](#)".



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 Anwendungsklustern zu vermeiden. Anweisungen finden Sie unter "[Die Produktdokumentation zu VMware NSX-T Data Center](#)".

Cluster-Anforderungen für Google Anthos

Wenn Sie Astra Control Center auf einem Google Anthos Cluster hosten, beachten Sie, dass Google Anthos standardmäßig den MetalLB Load Balancer und den Istio Ingress Gateway-Dienst enthält. So können Sie die generischen Ingress-Funktionen von Astra Control Center während der Installation einfach nutzen. Siehe "[Konfigurieren Sie Astra Control Center](#)" Entsprechende Details.

Unterstützte Storage-Back-Ends

Astra Control Center unterstützt folgende Storage-Back-Ends.

- NetApp ONTAP 9.5 oder neuere AFF und FAS Systeme
- NetApp ONTAP 9.8 oder neuere AFF und FAS Systeme für SnapMirror-basierte Applikationsreplizierung
- NetApp Cloud Volumes ONTAP

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

Sie können überprüfen, ob Ihr ONTAP System über die erforderlichen Lizenzen verfügt. Siehe "[Managen Sie ONTAP Lizenzen](#)".

Anforderungen für Applikationscluster

Astra Control Center hat folgende Anforderungen für Cluster, die Sie über das Astra Control Center verwalten möchten. Diese Anforderungen gelten auch, wenn der zu verwaltende Cluster der Betriebsumgebung ist, der das Astra Control Center hostet.

- Die neueste Version von Kubernetes "[snapshot-Controller-Komponente](#)" Installiert ist
- Astra Trident "[Objekt der Volumesnapshotklasse](#)" Wurde von einem Administrator definiert
- Im Cluster ist eine standardmäßige Kubernetes-Storage-Klasse vorhanden
- Mindestens eine Storage-Klasse ist für die Verwendung von Astra Trident konfiguriert



Ihr Applikations-Cluster sollte einen haben `kubeconfig.yaml` Datei, die nur ein `context`-Element definiert. In der Kubernetes-Dokumentation für finden Sie "[Informationen zum Erstellen von kubeconfig-Dateien](#)".



Wenn Sie Anwendungscluster in einer Rancher-Umgebung verwalten, ändern Sie den Standardkontext des Anwendungsclusters im `kubeconfig` Die von Rancher bereitgestellte Datei verwendet anstelle des Rancher API-Serverkontexts einen Steuerebenen-Kontext. So wird die Last auf dem Rancher API Server reduziert und die Performance verbessert.

Anforderungen für das Applikationsmanagement

Astra Control verfügt über folgende Anforderungen an das Applikationsmanagement:

- **Lizenzierung:** Zur Verwaltung von Anwendungen mit dem Astra Control Center benötigen Sie eine Astra Control Center-Lizenz.
- **Namespaces:** Astra Control erfordert, dass eine App nicht mehr als einen Single Namespace umfasst, aber ein Namespace kann mehr als eine App enthalten.
- **StorageClass:** Wenn Sie eine Anwendung mit einem explizit eingestellten StorageClass installieren und die App klonen müssen, muss das Zielcluster für den Klonvorgang die ursprünglich angegebene StorageClass haben. Das Klonen einer Applikation, deren StorageClass explizit auf ein Cluster festgelegt ist, das nicht über dieselbe StorageClass verfügt, schlägt fehl.
- **Kubernetes-Ressourcen:** Applikationen, die nicht mit Astra Control gesammelte Kubernetes-Ressourcen verwenden, verfügen unter Umständen nicht über umfassende Funktionen zum App-Datenmanagement. Astra Control sammelt die folgenden Kubernetes-Ressourcen:

ClusterCole	ClusterrollenBding	Konfigmap
Kronjob	KundenressourcenDefinition	Benutzerressource
DemonSet	BereitstellungConfig	Horizon PodAutoscaler
Eindringen	MutatingWebhook	Netzwerkrichtlinie
PersistentVolumeClaim	Pod	PodDisruptionBudget
PodTemplate	ReplicaSet	Rolle
Rollenverschwarten	Route	Geheim
Service	Service Account	StatfulSet
ValidierenWebhook		

Replikationsvoraussetzungen

Die Astra Control Applikationsreplizierung erfordert, dass die folgenden Voraussetzungen erfüllt sein müssen, bevor Sie beginnen:

- Um eine nahtlose Disaster Recovery zu erreichen, empfehlen wir Ihnen, Astra Control Center in einer dritten Fehlerdomäne oder an einem sekundären Standort einzusetzen.
- Das Kubernetes-Cluster der Applikation und ein Kubernetes Ziel-Cluster müssen verfügbar und mit zwei ONTAP Clustern verbunden sein, im Idealfall für unterschiedliche Ausfall-Domains oder Standorte.

- ONTAP-Cluster und die Host-SVM müssen gekoppelt sein. Siehe "[Übersicht über Cluster- und SVM-Peering](#)".
- Die gekoppelte Remote SVM muss für Trident auf dem Ziel-Cluster verfügbar sein.
- Trident Version 22.07 oder höher muss sowohl auf den Quell- als auch Ziel-ONTAP Clustern vorhanden sein.
- Asynchrone ONTAP SnapMirror Lizenzen mit dem Datensicherungs-Bundle müssen sowohl auf den Quell- als auch auf den Ziel-ONTAP Clustern aktiviert werden. Siehe "[Übersicht über die SnapMirror Lizenzierung in ONTAP](#)".
- Wenn Sie dem Astra Control Center ein ONTAP-Speicher-Backend hinzufügen, wenden Sie die Benutzeranmeldeinformationen auf die Rolle „Admin“ an, die über Zugriffsmethoden verfügt `http` und `ontapi` auf beiden ONTAP Clustern aktiviert. Siehe "[Benutzerkonten Verwalten](#)". Finden Sie weitere Informationen.
- Sowohl Quell- als auch Ziel-Kubernetes-Cluster als auch ONTAP-Cluster müssen von Astra Control gemanagt werden.



Sie können gleichzeitig eine andere Applikation (auf dem anderen Cluster oder Standort ausgeführt) in die entgegengesetzte Richtung replizieren. So können beispielsweise Applikationen A, B und C von Datacenter 1 nach Datacenter 2 repliziert werden. Applikationen X, Y und Z können von Datacenter 2 zu Datacenter 1 repliziert werden.

Erfahren Sie, wie Sie "[Replizieren von Applikationen auf einem Remote-System mit SnapMirror Technologie](#)".

Unterstützte Installationsmethoden für Anwendungen

Astra Control unterstützt folgende Installationsmethoden für Anwendungen:

- **Manifest-Datei:** Astra Control unterstützt Apps, die aus einer Manifest-Datei mit `kubectl` installiert wurden. Beispiel:

```
kubectl apply -f myapp.yaml
```

- **Helm 3:** Wenn Sie Helm zur Installation von Apps verwenden, benötigt Astra Control Helm Version 3. Das Management und Klonen von Apps, die mit Helm 3 installiert sind (oder ein Upgrade von Helm 2 auf Helm 3), wird vollständig unterstützt. Das Verwalten von mit Helm 2 installierten Apps wird nicht unterstützt.
- **Vom Betreiber bereitgestellte Apps:** Astra Control unterstützt Apps, die mit Betreibern mit Namespace-Scoped installiert sind. Im Folgenden sind einige Apps aufgeführt, die für dieses Installationsmodell validiert wurden:
 - "[Apache K8ssandra](#)"
 - "[Jenkins CI](#)"
 - "[Percona XtraDB Cluster](#)"



Ein Operator und die von ihm zu installieren App müssen denselben Namespace verwenden. Möglicherweise müssen Sie die yaml-Bereitstellungsdatei ändern, um sicherzustellen, dass dies der Fall ist.

Zugang zum Internet

Sie sollten feststellen, ob Sie einen externen Zugang zum Internet 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"](#).

Lizenz

Astra Control Center erfordert eine Astra Control Center-Lizenz für die volle Funktionalität. Anfordern einer Evaluierungslizenz oder Volllizenz von NetApp. Sie benötigen eine Lizenz zum Schutz Ihrer Applikationen und Daten. Siehe ["Funktionen des Astra Control Center"](#) Entsprechende Details.

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

Details zu Lizenzen, die für ONTAP Storage Back-Ends erforderlich sind, finden Sie unter ["Unterstützte Storage-Back-Ends"](#).

Weitere Informationen zur Funktionsweise von Lizenzen finden Sie unter ["Lizenzierung"](#).

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 mit dem MetalLB Load Balancer dem Dienst automatisch eine externe IP-Adresse zuweisen. In der Konfiguration des internen DNS-Servers sollten Sie den ausgewählten DNS-Namen für Astra Control Center auf die Load-Balanced IP-Adresse verweisen.



Wenn Sie Astra Control Center auf einem Tanzu Kubernetes Grid Cluster hosten, nutzen Sie den `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.

Weitere Informationen finden Sie unter ["Eindringen für den Lastenausgleich einrichten"](#).

Netzwerkanforderungen

Die Betriebsumgebung, die als Host für Astra Control Center fungiert, 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	443	HTTPS	Cloud Insights Kommunikation
Astra Control Center	Amazon S3 Storage-Bucket-Provider	443	HTTPS	Amazon S3 Storage-Kommunikation
Astra Control Center	NetApp AutoSupport	443	HTTPS	Kommunikation zwischen NetApp AutoSupport

Unterstützte Webbrowser

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

Wie es weiter geht

Sehen Sie sich die an ["Schnellstart"](#) Überblick.

Schnellstart für Astra Control Center

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

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

1

Kubernetes-Cluster-Anforderungen prüfen

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

Erfahren Sie mehr über das ["Anforderungen des Astra Control Centers"](#).

2

Laden Sie Astra Control Center herunter und installieren Sie es

- Laden Sie das Astra Control Center von der herunter ["NetApp Support Site Astra Control Center Download-Seite"](#).
- Installieren Sie Astra Control Center in Ihrer lokalen Umgebung.

Optional können Sie Astra Control Center mit Red hat OperatorHub installieren.

Optional kann Astra Control Center mit einem Cloud Volumes ONTAP Storage-Backend installiert werden.

Weitere Informationen zu ["Installieren des Astra Control Center"](#).

3

Führen Sie einige erste Setup-Aufgaben aus

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

Erfahren Sie mehr über das ["Initialer Einrichtungsvorgang"](#).

4

Nutzen Sie Das Astra Control Center

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

- Eine App verwalten. Erfahren Sie mehr darüber, wie Sie ["Applikationsmanagement"](#).
- Schützen Sie Applikationen durch die Konfiguration von Sicherheitsrichtlinien für Applikationen, das Replizieren von Applikationen auf Remote-Systeme und das Klonen und Migrieren von Applikationen. Erfahren Sie mehr darüber, wie Sie ["Schützen von Applikationen"](#).
- Konten verwalten (einschließlich Benutzer, Rollen, LDAP zur Benutzerauthentifizierung, Anmeldedaten, Repository-Verbindungen usw.). Erfahren Sie mehr darüber, wie Sie ["Benutzer managen"](#).
- Optional können Sie eine Verbindung zu NetApp Cloud Insights herstellen, um Kennzahlen zum Zustand von System, Kapazität und Durchsatz innerhalb der Astra Control Center UI anzuzeigen. Weitere Informationen zu ["Verbindung zu Cloud Insights wird hergestellt"](#).

Fahren Sie mit dieser Schnellstartanleitung fort

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

Weitere Informationen

- ["Verwenden Sie die Astra Control API"](#)

Übersicht über die Installation

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

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

Installieren Sie das Astra Control Center mithilfe des Standardprozesses

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

Für Red hat OpenShift-Umgebungen können Sie ein verwenden ["Alternativverfahren"](#) So installieren Sie Astra Control Center mithilfe des OpenShift OperatorHub.

Was Sie benötigen

- ["Bevor Sie mit der Installation beginnen, bereiten Sie Ihre Umgebung auf die Implementierung des Astra Control Center vor"](#).
- 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 Cluster Operator in einem ordnungsgemäßen Zustand und verfügbar sind.

```
kubectl get clusteroperators
```

- 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 routungsfähig 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 seinen eigenen Cert-Manager installiert.

Über diese Aufgabe

Der Astra Control Center-Installationsprozess führt Folgendes aus:

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



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



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



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

Schritte

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

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

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

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

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

4. Extrahieren Sie die Bilder:

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

Installieren Sie das NetApp Astra kubectl Plug-in

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

Was Sie benötigen

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

Schritte

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

```
ls kubectl-astra/
```

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

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

Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

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

Docker

1. Wechseln Sie in das Astra-Verzeichnis:

```
cd acc
```

2. Schieben Sie die Paketbilder im Astra Control Center-Bildverzeichnis in Ihre lokale Registrierung. Führen Sie folgende Ersetzungen durch, bevor Sie den Befehl ausführen:
 - ERSETZEN SIE DIE BUNDLE_FILE durch den Namen der Astra Control Bundle-Datei (z. B. `acc.manifest.yaml`).
 - ERSETZEN SIE MY_REGISTRY durch die URL des Docker Repositorys.
 - 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_REGISTRY  
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

Podman

1. Melden Sie sich bei Ihrer Registrierung an:

```
podman login [your_registry_path]
```

2. Führen Sie das folgende Skript aus und machen Sie die Substitution <YOUR_REGISTRY> wie in den Kommentaren angegeben:

```

# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //'')

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done

```

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]
```

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/22.08.1-26`).

```
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, müssen Sie das Geheimnis für den Namespace neu generieren, nachdem der Namespace neu erstellt wurde.

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

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

Beispielantwort:

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

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

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

Antwort

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


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

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

Installieren Sie den Operator Astra Control Center

1. 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](#).
- d. (Für Installationen mit Astra Data Store Vorschau) Siehe dieses bekannte Problem bzgl. "[Provisorer der Speicherklasse und zusätzliche Änderungen, die Sie an der YAML vornehmen müssen](#)".

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
    name: acc-operator-controller-manager
    namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
        - args:
          - --secure-listen-address=0.0.0.0:8443
          - --upstream=http://127.0.0.1:8080/
          - --logtostderr=true
          - --v=10
          image: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
          - --health-probe-bind-address=:8081
          - --metrics-bind-address=127.0.0.1:8080
          - --leader-elect
          command:
            - /manager
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
          image: [your_registry_path]/acc-operator:[version x.y.z]
          imagePullPolicy: IfNotPresent
      imagePullSecrets: []

```

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_min.yaml) Um Konto, AutoSupport, Registrierung und andere notwendige Konfigurationen zu machen:



astra_control_center_min.yaml Ist die Standard-CR und ist für die meisten Installationen geeignet. Machen Sie sich mit allen vertraut ["CR-Optionen und ihre potenziellen Werte"](#) Damit Sie Astra Control Center richtig für Ihre Umgebung einsetzen können. Falls für Ihre Umgebung zusätzliche Anpassungen erforderlich sind, können Sie dies verwenden astra_control_center.yaml Als Alternative CR.



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

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

- b. Ändern Sie das `accountName` Zeichenfolge an den Namen, den Sie dem Konto zuordnen möchten.
- c. Ändern Sie das `astraAddress` Zeichenfolge an den FQDN, den Sie in Ihrem Browser für den Zugriff auf Astra verwenden möchten. Verwenden Sie es nicht `http://` Oder `https://` In der Adresse. Kopieren Sie diesen FQDN zur Verwendung in einem [Später Schritt](#).
- d. Ändern Sie das `email` Zeichenfolge zur standardmäßigen ursprünglichen Administratoradresse. Kopieren Sie diese E-Mail-Adresse zur Verwendung in A [Später Schritt](#).
- e. Ändern `enrolled` Für AutoSupport bis `false` Für Websites ohne Internetverbindung oder Aufbewahrung `true` Für verbundene Standorte.
- f. Wenn Sie einen externen Zertifikaten-Manager verwenden, fügen Sie folgende Zeilen zu hinzu `spec`:

```
spec:
  crds:
    externalCertManager: true
```

- g. (Optional) Geben Sie einen Vornamen ein `firstName` Und Nachname `lastName` Des Benutzers, der dem Konto zugeordnet ist. Sie können diesen Schritt jetzt oder später in der Benutzeroberfläche ausführen.
- h. (Optional) Ändern Sie den `storageClass` Nutzen Sie bei Bedarf für Ihre Installation einen anderen Trident Storage Class-Mitarbeiter.
- i. (Optional) Wenn der Cluster nach der Installation automatisch von Astra Control Center verwaltet werden soll und schon vorhanden ist [Schuf das Geheimnis, das den kubeconfig für diesen Cluster enthält](#)Geben Sie den Namen des Geheimnisses an, indem Sie dieser YAML-Datei ein neues Feld hinzufügen `astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"`
- j. Führen Sie einen der folgenden Schritte aus:

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

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

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

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

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



Einzelheiten zum Servicetyp von „loadbalancer“ und Ingress finden Sie unter ["Anforderungen"](#).

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

Komplette Astra Control Center und Bedienerinstallation

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

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

Beispielantwort:

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

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

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

Beispielantwort:

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

Überprüfen Sie den Systemstatus



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

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

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

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

Beispielantwort

NAME	READY	STATUS	RESTARTS
AGE			
acc-helm-repo-6b44d68d94-d8m55 13m	1/1	Running	0
activity-78f99ddf8-hltct 10m	1/1	Running	0
api-token-authentication-457nl 9m28s	1/1	Running	0
api-token-authentication-dgwsz 9m28s	1/1	Running	0
api-token-authentication-hmqqc 9m28s	1/1	Running	0
asup-75fd554dc6-m6qzh 9m38s	1/1	Running	0
authentication-6779b4c85d-92gds 8m11s	1/1	Running	0
bucket-service-7cc767f8f8-lqwr8 9m31s	1/1	Running	0
certificates-549fd5d6cb-5kmd6 9m56s	1/1	Running	0
certificates-549fd5d6cb-bkjh9 9m56s	1/1	Running	0
cloud-extension-7bcb7948b-hn8h2 10m	1/1	Running	0
cloud-insights-service-56ccf86647-fgg69 9m46s	1/1	Running	0
composite-compute-677685b9bb-7vgsf 10m	1/1	Running	0
composite-volume-657d6c5585-dnq79 9m49s	1/1	Running	0
credentials-755fd867c8-vrlmt 11m	1/1	Running	0
entitlement-86495cdf5b-nwhh2 10m	1/1	Running	2
features-5684fb8b56-8d6s8 10m	1/1	Running	0
fluent-bit-ds-rhx7v 7m48s	1/1	Running	0
fluent-bit-ds-rjms4 7m48s	1/1	Running	0
fluent-bit-ds-zf5ph 7m48s	1/1	Running	0
graphql-server-66d895f544-w6hjd 3m29s	1/1	Running	0

identity-744df448d5-rlcmm	1/1	Running	0
10m			
influxdb2-0	1/1	Running	0
13m			
keycloak-operator-75c965cc54-z7csw	1/1	Running	0
8m16s			
krakend-798d6df96f-9z2sk	1/1	Running	0
3m26s			
license-5fb7d75765-f8mjg	1/1	Running	0
9m50s			
login-ui-7d5b7df85d-l2s7s	1/1	Running	0
3m20s			
loki-0	1/1	Running	0
13m			
metrics-facade-599b9d7fcc-gtmgl	1/1	Running	0
9m40s			
monitoring-operator-67cc74f844-cdplp	2/2	Running	0
8m11s			
nats-0	1/1	Running	0
13m			
nats-1	1/1	Running	0
13m			
nats-2	1/1	Running	0
12m			
nautilus-769f5b74cd-k5jxm	1/1	Running	0
9m42s			
nautilus-769f5b74cd-kd9gd	1/1	Running	0
8m59s			
openapi-84f6ccd8ff-76kvp	1/1	Running	0
9m34s			
packages-6f59fc67dc-4g2f5	1/1	Running	0
9m52s			
polaris-consul-consul-server-0	1/1	Running	0
13m			
polaris-consul-consul-server-1	1/1	Running	0
13m			
polaris-consul-consul-server-2	1/1	Running	0
13m			
polaris-keycloak-0	1/1	Running	0
8m7s			
polaris-keycloak-1	1/1	Running	0
5m49s			
polaris-keycloak-2	1/1	Running	0
5m15s			
polaris-keycloak-db-0	1/1	Running	0
8m6s			

polaris-keycloak-db-1	1/1	Running	0
5m49s			
polaris-keycloak-db-2	1/1	Running	0
4m57s			
polaris-mongodb-0	2/2	Running	0
13m			
polaris-mongodb-1	2/2	Running	0
12m			
polaris-mongodb-2	2/2	Running	0
12m			
polaris-ui-565f56bf7b-zwr8b	1/1	Running	0
3m19s			
polaris-vault-0	1/1	Running	0
13m			
polaris-vault-1	1/1	Running	0
13m			
polaris-vault-2	1/1	Running	0
13m			
public-metrics-6d86d66444-2wbz1	1/1	Running	0
9m30s			
storage-backend-metrics-77c5d98dcd-dbhg5	1/1	Running	0
9m44s			
storage-provider-78c885f57c-6zcv4	1/1	Running	0
9m36s			
telegraf-ds-212m9	1/1	Running	0
7m48s			
telegraf-ds-qfzgh	1/1	Running	0
7m48s			
telegraf-ds-shrms	1/1	Running	0
7m48s			
telegraf-rs-bjpkt	1/1	Running	0
7m48s			
telemetry-service-6684696c64-qzfdf	1/1	Running	0
10m			
tenancy-6596b6c54d-vmppm	1/1	Running	0
10m			
traefik-7489dc59f9-6mnst	1/1	Running	0
3m19s			
traefik-7489dc59f9-xrkkg	1/1	Running	0
3m4s			
trident-svc-6c8dc458f5-jswcl	1/1	Running	0
10m			
vault-controller-6b954f9b76-gz9nm	1/1	Running	0
11m			

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

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



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

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

```
kubectl get AstraControlCenter -n netapp-acc
```

Antwort:

NAME	UUID	VERSION	ADDRESS
READY			
astra	ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	22.08.1-26	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, wie etwa den Lastausgleich in einem Cluster, managt.

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



Wenn Sie keinen Ingress-Controller einrichten möchten, können Sie ihn einstellen (`ingressType:AccTraefik`). Astra Control Center nutzt einen Service vom Typ „loadbalancer“ (`svc/traefik` Im Astra Control Center Namespace) und erfordert, dass ihm eine zugängliche externe IP-Adresse zugewiesen wird. Wenn in Ihrer Umgebung Load Balancer zugelassen sind und Sie noch nicht eine konfiguriert haben, können Sie MetallB oder einen anderen externen Service Load Balancer verwenden, um dem Dienst eine externe IP-Adresse zuzuweisen. In der Konfiguration des internen DNS-Servers sollten Sie den ausgewählten DNS-Namen für Astra Control Center auf die Load-Balanced IP-Adresse verweisen. Einzelheiten zum Servicetyp von „loadbalancer“ und Ingress finden Sie unter ["Anforderungen"](#).

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

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

Was Sie benötigen

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

Schritte für 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 in `netapp-acc` (Oder Custom-Name) Namespace mit entweder dem `v1beta1` (veraltet in Kubernetes Version weniger als oder 1.22) oder `v1` Ressourcentyp für entweder ein deprecated oder ein neues Schema:

Ausgabe:

```
apiVersion: networking.k8s.io/v1beta1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress
  namespace: istio-system
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          serviceName: traefik
          servicePort: 80
```

Für das neue Schema v1 gehen Sie wie folgt vor:

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

Ausgabe:

```

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: istio-system
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. Implementieren Sie wie gewohnt Astra Control Center.
6. Überprüfen Sie den Status des Eingangs:

```
kubectl get ingress -n netapp-acc
```

Antwort:

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

Schritte für Nginx Ingress Controller

1. Erstellen Sie ein Geheimnis des Typs [kubernetes.io/tls] Für einen privaten TLS-Schlüssel und ein Zertifikat in netapp-acc (Oder Custom-Name) Namespace wie in beschrieben "TLS-Geheimnisse".

2. Bereitstellung einer Ingress-Ressource in `netapp-acc` (Oder Custom-Name) Namespace mit entweder dem `v1beta1` (Veraltet in Kubernetes Version kleiner als oder 1.22) oder `v1` Ressourcentyp für ein deprecated oder ein neues Schema:
- a. Für A `v1beta1` Veraltete Schemas, folgen Sie diesem Beispiel:

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

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

```

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

```

Schritte für OpenShift-Eingangs-Controller

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

```

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

```

Melden Sie sich in der UI des Astra Control Center an

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

Schritte

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



Sie können nach der Anmeldung ein benutzerdefiniertes Zertifikat erstellen.

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



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

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



Wenn es sich um Ihre erste Anmeldung handelt und Sie das Passwort vergessen haben und noch keine anderen Administratorkonten erstellt wurden, wenden Sie sich an den NetApp Support, um Unterstützung bei der Passwortwiederherstellung zu erhalten.

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

Beheben Sie die Fehlerbehebung für die Installation

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

Schritte

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

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

Wie es weiter geht

Führen Sie die Implementierung durch ["Setup-Aufgaben"](#).

=

:allow-uri-read:

Einschränkungen der POD-Sicherheitsrichtlinie verstehen

Astra Control Center unterstützt die Einschränkung von Berechtigungen durch POD-Sicherheitsrichtlinien (PSPs). Mithilfe der POD-Sicherheitsrichtlinien können Sie begrenzen, welche Benutzer oder Gruppen Container ausführen können und welche Berechtigungen diese Container haben können.

Einige Kubernetes Distributionen, wie z. B. RKE2, verfügen über eine Standard-Pod-Sicherheitsrichtlinie, die zu restriktiv ist und bei der Installation von Astra Control Center Probleme verursacht.

Anhand der hier enthaltenen Informationen und Beispiele können Sie die von Astra Control Center erstellten POD-Sicherheitsrichtlinien verstehen und die Richtlinien für die POD-Sicherheit konfigurieren, die den erforderlichen Schutz bieten, ohne die Funktionen des Astra Control Center zu beeinträchtigen.

PSPs, die vom Astra Control Center installiert werden

Astra Control Center erstellt während der Installation mehrere POD-Sicherheitsrichtlinien. Einige davon sind dauerhaft, und einige von ihnen werden während bestimmter Operationen erstellt und werden entfernt, sobald der Vorgang abgeschlossen ist.

PSPs, die während der Installation erstellt wurden

Bei der Installation von Astra Control Center installiert der Astra Control Center-Operator eine benutzerdefinierte POD-Sicherheitsrichtlinie, ein Rollenobjekt und ein rollenbindendes Objekt, um die Implementierung von Astra Control Center-Diensten im Astra Control Center-Namespace zu unterstützen.

Die neue Richtlinie und die neuen Objekte haben folgende Attribute:

```
kubectl get psp
```

NAME	PRIV	CAPS	SELINUX	RUNASUSER
FSGROUP SUPGROUP READONLYROOTFS		VOLUMES		
avp-ppsp	false		RunAsAny	RunAsAny
RunAsAny RunAsAny	false	*		
netapp-astra-deployment-ppsp	false		RunAsAny	RunAsAny
RunAsAny RunAsAny	false	*		

```
kubectl get role
```

NAME	CREATED AT
netapp-astra-deployment-role	2022-06-27T19:34:58Z

```
kubectl get rolebinding
```

NAME	ROLE
AGE	
netapp-astra-deployment-rb	Role/netapp-astra-deployment-role
32m	

Während des Backup-Betriebs erstellte PSPs

Astra Control Center erstellt während des Backup-Betriebs eine dynamische Pod-Sicherheitsrichtlinie, ein ClusterRollenobjekt und ein rollenbindendes Objekt. Diese unterstützen den Backup-Prozess, der in einem separaten Namespace geschieht.

Die neue Richtlinie und die neuen Objekte haben folgende Attribute:

```
kubectl get psp
```

NAME	SELINUX	RUNASUSER	PRIV	FSGROUP	CAPS	SUPGROUP	READONLYROOTFS	VOLUMES
netapp-astra-backup			false		DAC_READ_SEARCH			
RunAsAny	RunAsAny	RunAsAny	RunAsAny	RunAsAny		false		*

```
kubectl get role
```

NAME	CREATED AT
netapp-astra-backup	2022-07-21T00:00:00Z

```
kubectl get rolebinding
```

NAME	ROLE	AGE
netapp-astra-backup	Role/netapp-astra-backup	62s

PSPs, die während des Clustermanagements erstellt wurden

Wenn Sie einen Cluster verwalten, installiert Astra Control Center den netapp Monitoring Operator im Managed Cluster. Dieser Operator erstellt eine Pod-Sicherheitsrichtlinie, ein ClusterRole-Objekt und ein RoleBinding-Objekt, um Telemetrieservices im Namespace von Astra Control Center bereitzustellen.

Die neue Richtlinie und die neuen Objekte haben folgende Attribute:

```
kubectl get psp
```

NAME	SELINUX	RUNASUSER	PRIV	FSGROUP	CAPS	SUPGROUP	READONLYROOTFS	VOLUMES
netapp-monitoring-bsp-nkmo			true		AUDIT_WRITE,NET_ADMIN,NET_RAW			
RunAsAny	RunAsAny	RunAsAny	RunAsAny	RunAsAny		false		*

```
kubectl get role
```

NAME	CREATED AT
netapp-monitoring-role-privileged	2022-07-21T00:00:00Z

```
kubectl get rolebinding
```

NAME	ROLE	AGE
netapp-monitoring-role-binding-privileged	Role/netapp-monitoring-role-privileged	2m5s

Aktivieren der Netzwerkkommunikation zwischen Namespaces

Einige Umgebungen verwenden NetworkPolicy-Konstrukte, um den Datenverkehr zwischen Namespaces zu beschränken. Der Astra Control Center Operator, Astra Control Center und das Astra Plugin für VMware vSphere sind allesamt 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
```

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
```

3. Wenden Sie die folgende Ressourcendatei an, um das ACC-avp-Netzwerk-Policy-Objekt zu konfigurieren, damit das Astra Plugin für VMware vSphere Services Anfragen an die Astra Control Center Services stellen kann. 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 FOR VMWARE VSPHERE NAMESPACE NAME
```

4. Wenden Sie die folgende Ressourcendatei an, um das ACC-Operator-Network-Policy-Objekt so zu konfigurieren, dass der Astra Control Center-Operator mit den Astra Control Center-Diensten kommunizieren kann. 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

```

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. Sie müssen sie 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. Holen Sie sich die Ressourcenkontingente im netapp-ACC Namespace:

```
kubectl get quota -n netapp-acc
```

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
```

```
kubectl delete resourcequota pods-low -n netapp-acc
```

```
kubectl delete resourcequota pods-medium -n netapp-acc
```

3. Grenzbereiche im netapp-ACC Namespace abrufen:

```
kubectl get limits -n netapp-acc
```

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
```

=

:allow-uri-read:

Installieren Sie Astra Control Center mit OpenShift OperatorHub

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

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

Was Sie benötigen

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

```
oc get clusteroperators
```

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

sind (available Ist true):

```
oc get apiservices
```

- Erstellen Sie in Ihrem Rechenzentrum eine FQDN-Adresse für Astra Control Center.
- Erhalten Sie die erforderlichen Berechtigungen und den Zugriff auf die Red hat OpenShift Container Platform, um die beschriebenen Installationsschritte durchzuführen.
- Wenn bereits ein Zertifikat-Manager im Cluster vorhanden ist, müssen Sie einen Teil durchführen "[Erforderliche Schritte](#)" Damit Astra Control Center nicht seinen eigenen Cert-Manager installiert.

Schritte

- [Laden Sie das Astra Control Center Bundle herunter und entpacken Sie es aus](#)
- [Installieren Sie das NetApp Astra kubectl Plug-in](#)
- [Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu](#)
- [Suchen Sie die Installationsseite des Bedieners](#)
- [Installieren Sie den Operator](#)
- [Installieren Sie Astra Control Center](#)

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

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

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

4. Extrahieren Sie die Bilder:

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

Installieren Sie das NetApp Astra kubectl Plug-in

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

Was Sie benötigen

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

Informationen.

Schritte

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

```
ls kubectl-astra/
```

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

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

Fügen Sie die Bilder Ihrer lokalen Registrierung hinzu

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

Docker

1. Wechseln Sie in das Astra-Verzeichnis:

```
cd acc
```

2. Schieben Sie die Paketbilder im Astra Control Center-Bildverzeichnis in Ihre lokale Registrierung. Führen Sie folgende Ersetzungen durch, bevor Sie den Befehl ausführen:
 - ERSETZEN SIE DIE BUNDLE_FILE durch den Namen der Astra Control Bundle-Datei (z. B. `acc.manifest.yaml`).
 - ERSETZEN SIE MY_REGISTRY durch die URL des Docker Repositorys.
 - 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_REGISTRY  
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

Podman

1. Melden Sie sich bei Ihrer Registrierung an:

```
podman login [your_registry_path]
```

2. Führen Sie das folgende Skript aus und machen Sie die Substitution <YOUR_REGISTRY> wie in den Kommentaren angegeben:


```

# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s) : //' )

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

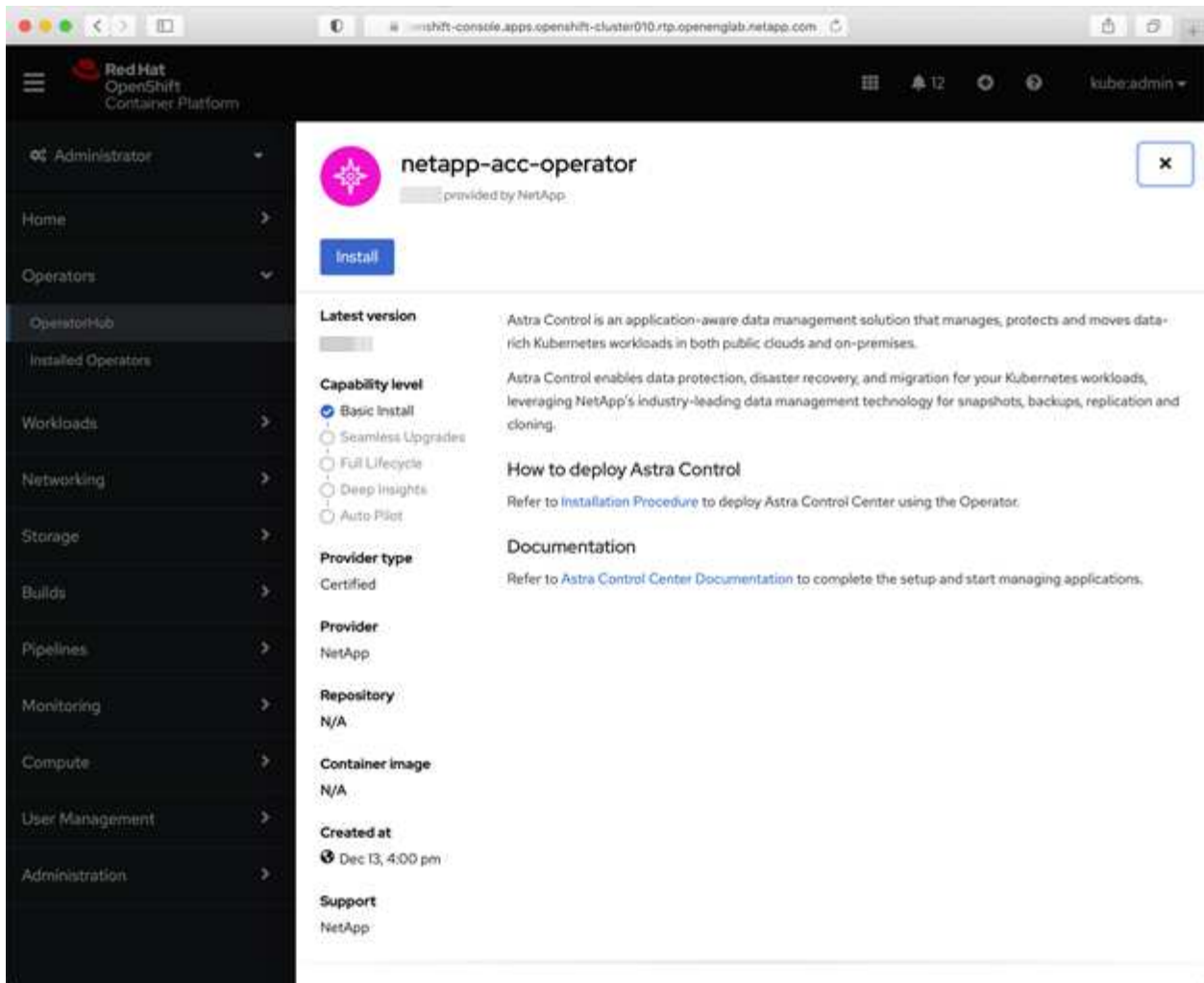
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done

```

Suchen Sie die Installationsseite des Bedieners

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



- i. Melden Sie sich in der OpenShift Container Platform UI an.
 - ii. Wählen Sie im Seitenmenü die Option **Operatoren > OperatorHub** aus.
 - iii. Wählen Sie den Operator des NetApp Astra Control Center aus.
 - iv. Wählen Sie **Installieren**.
- Aus Dem Red Hat Ecosystem Catalog:

Red Hat Ecosystem Catalog | Hardware | Software | Cloud & service providers

Home > Software > OpenShift operators > Astra Control Center

Astra Control Center

Provided by [NetApp](#)

Application-aware data management built for OpenShift

[Deploy and use](#)

[Overview](#) | [Features & benefits](#) | [Documentation](#) | [Deploy & use](#) | [FAQs](#) | [Get support](#)

[Have feedback?](#)

Overview

- 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 in der Detailansicht des Bedieners Astra Control Center die Option aus `Create instance` Im Abschnitt über die bereitgestellten APIs.
2. Füllen Sie die aus `Create AstraControlCenter` Formularfeld:
 - a. Behalten Sie den Namen des Astra Control Center bei oder passen Sie diesen an.
 - b. (Optional) Aktivieren oder Deaktivieren von Auto Support. Es wird empfohlen, die Auto Support-Funktion beizubehalten.

- c. Geben Sie die Astra Control Center-Adresse ein. Kommen Sie nicht herein `http://` Oder `https://` In der Adresse.
 - d. Geben Sie die Astra Control Center-Version ein, z. B. 21.12.60.
 - e. Geben Sie einen Kontonamen, eine E-Mail-Adresse und einen Administratorknamen ein.
 - f. Beibehaltung der Standard-Richtlinie zur Rückgewinnung von Volumes
 - g. Geben Sie in **Image Registry** Ihren lokalen Container Image Registry-Pfad ein. Kommen Sie nicht herein `http://` Oder `https://` In der Adresse.
 - h. Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, geben Sie das Geheimnis ein.
 - i. Geben Sie den Vornamen des Administrators ein.
 - j. Konfiguration der Ressourcenskalisierung
 - k. Behalten Sie die Standard-Storage-Klasse bei.
 - l. Definieren Sie die Einstellungen für die Verarbeitung von CRD.
3. Wählen Sie `Create`.

Wie es weiter geht

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

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

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

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

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 Anforderungen an die Ressourcen entspricht, die in der offiziellen Dokumentation der Umgebung aufgeführt sind.

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

Komponente	Anforderungen
Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität	Mindestens 300 GB verfügbar
Worker-Nodes (AWS EC2 Anforderung)	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
Load Balancer	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
FQDN	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
Astra Trident (installiert als Teil der Kubernetes Cluster Discovery in NetApp Cloud Manager)	Astra Trident 21.04 oder höher ist installiert und konfiguriert und NetApp ONTAP Version 9.5 oder höher als Storage-Backend

Komponente	Anforderungen
Bildregistrierung	<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>
Konfiguration von Astra Trident/ONTAP	<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 Importieren des Kubernetes Clusters in NetApp Cloud Manager erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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



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

Überblick über die Implementierung für AWS

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

Jeder dieser Schritte wird unten im Detail erklärt.

1. [dass Sie über ausreichende IAM-Berechtigungen verfügen.](#)
2. [Installation eines RedHat OpenShift-Clusters in AWS.](#)
3. [Konfigurieren von AWS.](#)
4. [NetApp Cloud Manager konfigurieren.](#)

5. Installieren Sie Astra Control Center.

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

Stellen Sie sicher, dass die IAM-Rollen und -Berechtigungen ausreichend sind, damit ein RedHat OpenShift-Cluster und ein NetApp Cloud Manager Connector installiert werden können.

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

Installation eines RedHat OpenShift-Clusters in AWS

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

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

Konfigurieren von AWS

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

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

1. Virtuelles AWS Netzwerk erstellen.
2. EC2 Computing-Instanzen prüfen. Dabei können es sich um einen Bare Metal Server oder VMs in AWS 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



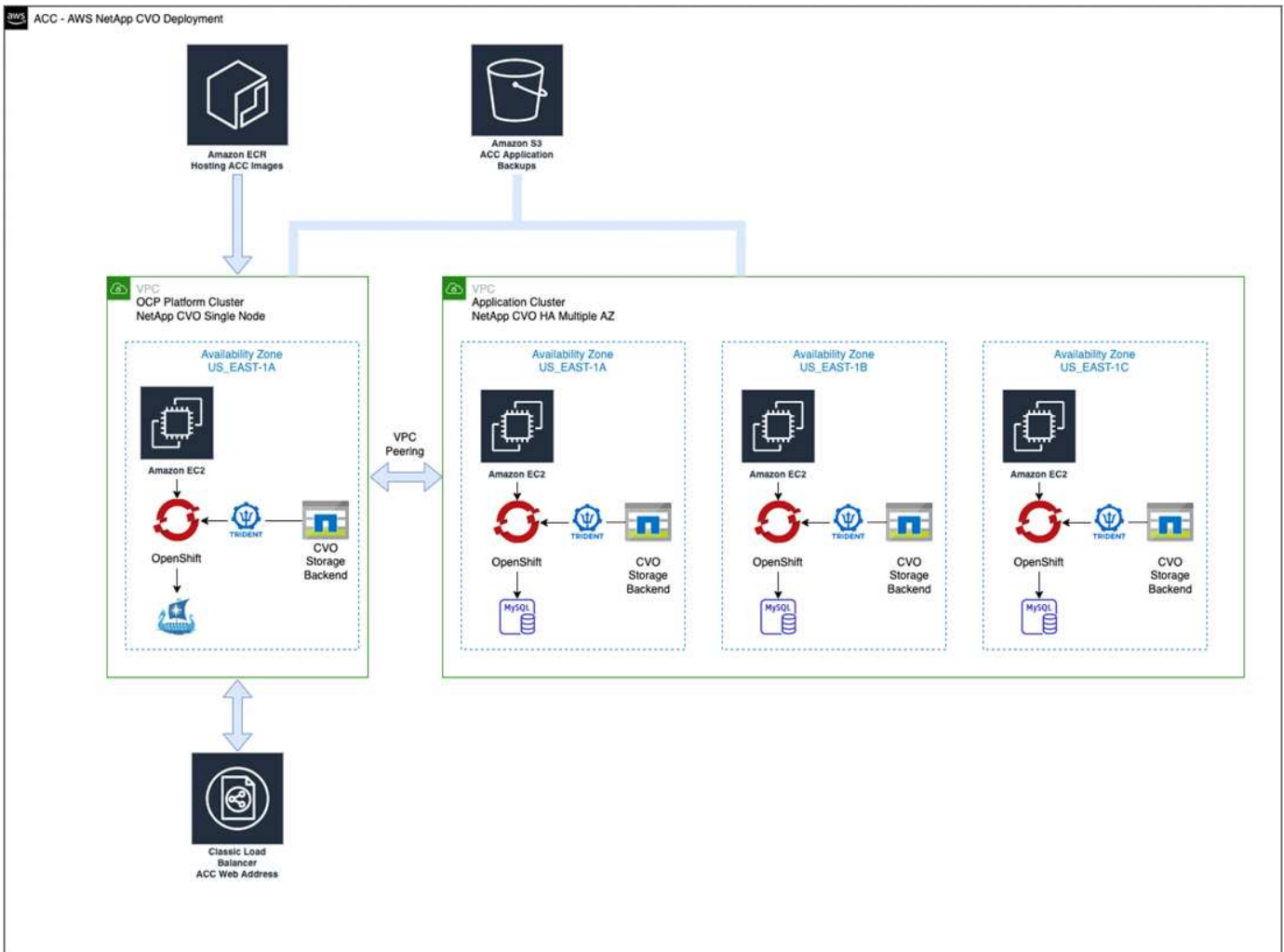
Wenn Sie den ECR nicht erstellen, kann Astra Control Center mit einem AWS Backend nicht auf die Monitoring-Daten von einem Cluster mit Cloud Volumes ONTAP zugreifen. Das Problem wird verursacht, wenn der Cluster, den Sie mit Astra Control Center ermitteln und verwalten möchten, keinen AWS ECR-Zugriff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



NetApp Cloud Manager konfigurieren

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

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

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

Schritte

1. Fügen Sie Ihre Zugangsdaten zu Cloud Manager hinzu.
2. Erstellen Sie einen Arbeitsbereich.
3. Fügen Sie einen Connector für AWS hinzu. Entscheiden Sie sich für AWS als Provider.
4. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
 - a. Ort: „Amazon Web Services (AWS)“
 - b. Typ: „Cloud Volumes ONTAP HA“
5. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.
 - a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s > Cluster list > Cluster-Details** wählen.

- b. Beachten Sie oben rechts die Trident-Version.
- c. Beachten Sie die Cloud Volumes ONTAP Cluster-Storage-Klassen, für die NetApp als Provisionierung angezeigt wird.

Dies importiert Ihr Red hat OpenShift-Cluster und weist ihm eine Standard-Speicherklasse zu. Sie wählen die Speicherklasse aus. Trident wird automatisch im Rahmen des Import- und Erkennungsvorgangs installiert.

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



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

Installieren Sie Astra Control Center

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



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 der Betriebsumgebung für GCP



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

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

Komponente	Anforderungen
Back-End NetApp Cloud Volumes ONTAP Storage-Kapazität	Mindestens 300 GB verfügbar
Worker-Nodes (GCP-Compute-Anforderung)	Insgesamt mindestens 3 Worker-Nodes mit 4 vCPU-Kernen und jeweils 12 GB RAM
Load Balancer	Der Servicetyp „loadbalancer“ ist für den Ingress Traffic verfügbar, der an Services im Cluster der Betriebsumgebung gesendet werden kann
FQDN (GCP-DNS-ZONE)	Eine Methode zum Zeigen des FQDN von Astra Control Center auf die Load Balanced IP-Adresse
Astra Trident (installiert als Teil der Kubernetes Cluster Discovery in NetApp Cloud Manager)	Astra Trident 21.04 oder höher ist installiert und konfiguriert und NetApp ONTAP Version 9.5 oder höher als Storage-Backend
Bildregistrierung	<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>
Konfiguration von Astra Trident/ONTAP	<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 Importieren des Kubernetes Clusters in NetApp Cloud Manager erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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. [Installation eines RedHat OpenShift-Clusters in GCP](#).
2. [Erstellung eines GCP-Projekts und einer virtuellen Private Cloud](#).
3. [dass Sie über ausreichende IAM-Berechtigungen verfügen](#).
4. [GCP konfigurieren](#).
5. [NetApp Cloud Manager konfigurieren](#).
6. [Installation und Konfiguration des Astra Control Center](#).

Installation eines RedHat OpenShift-Clusters in GCP

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

Anweisungen zur Installation finden Sie im folgenden Abschnitt:

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

Erstellung eines GCP-Projekts und einer virtuellen Private Cloud

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



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

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

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

Stellen Sie sicher, dass die IAM-Rollen und -Berechtigungen ausreichend sind, damit ein RedHat OpenShift-Cluster und ein NetApp Cloud Manager Connector installiert werden 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 Mindestanforderungen für Ressourcen von Astra 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 Cloud Manager konfigurieren

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

Folgen Sie der Dokumentation zum Cloud Manager, um die folgenden Schritte auszuführen. Siehe "[Erste Schritte mit Cloud Volumes ONTAP in GCP](#)".

Was Sie benötigen

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

Schritte

1. Fügen Sie Ihre Zugangsdaten zu Cloud Manager 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 Konnektors in GCP von Cloud Manager"](#).
 - c. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.
3. Schaffen Sie eine Arbeitsumgebung für Ihre Cloud-Umgebung.
 - a. Speicherort: „GCP“
 - b. Typ: „Cloud Volumes ONTAP HA“
4. Importieren Sie den OpenShift-Cluster. Der Cluster wird mit der gerade erstellten Arbeitsumgebung verbunden.
 - a. Zeigen Sie die NetApp Cluster-Details an, indem Sie **K8s > Cluster list > Cluster-Details** wählen.
 - b. Beachten Sie 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. Trident wird automatisch im Rahmen des Import- und Erkennungsvorgangs installiert.

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



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

Installieren Sie Astra Control Center

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 Anforderungen an die Ressourcen entspricht, die in der offiziellen Dokumentation der Umgebung aufgeführt sind.

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

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

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

Komponente	Anforderungen
Konfiguration von Astra Trident/ONTAP	<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 Importieren des Kubernetes Clusters in NetApp Cloud Manager erstellt werden. Die folgenden Aufgaben werden von Astra Trident bereitgestellt:</p> <ul style="list-style-type: none"> • vsaworkingenvironment-<code><></code>-ha-nas csi.trident.netapp.io • vsaworkingenvironment-<code><></code>-ha-san csi.trident.netapp.io • vsaworkingenvironment-<code><></code>-single-nas csi.trident.netapp.io • vsaworkingenvironment-<code><></code>-single-san csi.trident.netapp.io



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

Überblick über die Implementierung für Azure

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

Jeder dieser Schritte wird unten im Detail erklärt.

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

Installieren Sie einen RedHat OpenShift-Cluster auf Azure

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

Installationsanweisungen finden Sie in der Dokumentation zu RedHat auf "[Installation von OpenShift-Cluster auf Azure](#)" Und "[Installieren eines Azure-Kontos](#)".

Erstellen von Azure Ressourcengruppen

Erstellen Sie mindestens eine Azure-Ressourcengruppe.



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

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

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

Stellen Sie sicher, dass die IAM-Rollen und -Berechtigungen ausreichend sind, damit ein RedHat OpenShift-Cluster und ein NetApp Cloud Manager Connector installiert werden können.

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

Konfigurieren Sie Azure

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

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

1. Virtuelles Azure Netzwerk erstellen.
2. Prüfen Sie die Computing-Instanzen. Dabei können es sich um einen Bare Metal Server oder VMs in Azure 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.

NetApp Cloud Manager konfigurieren

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

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

Was Sie benötigen

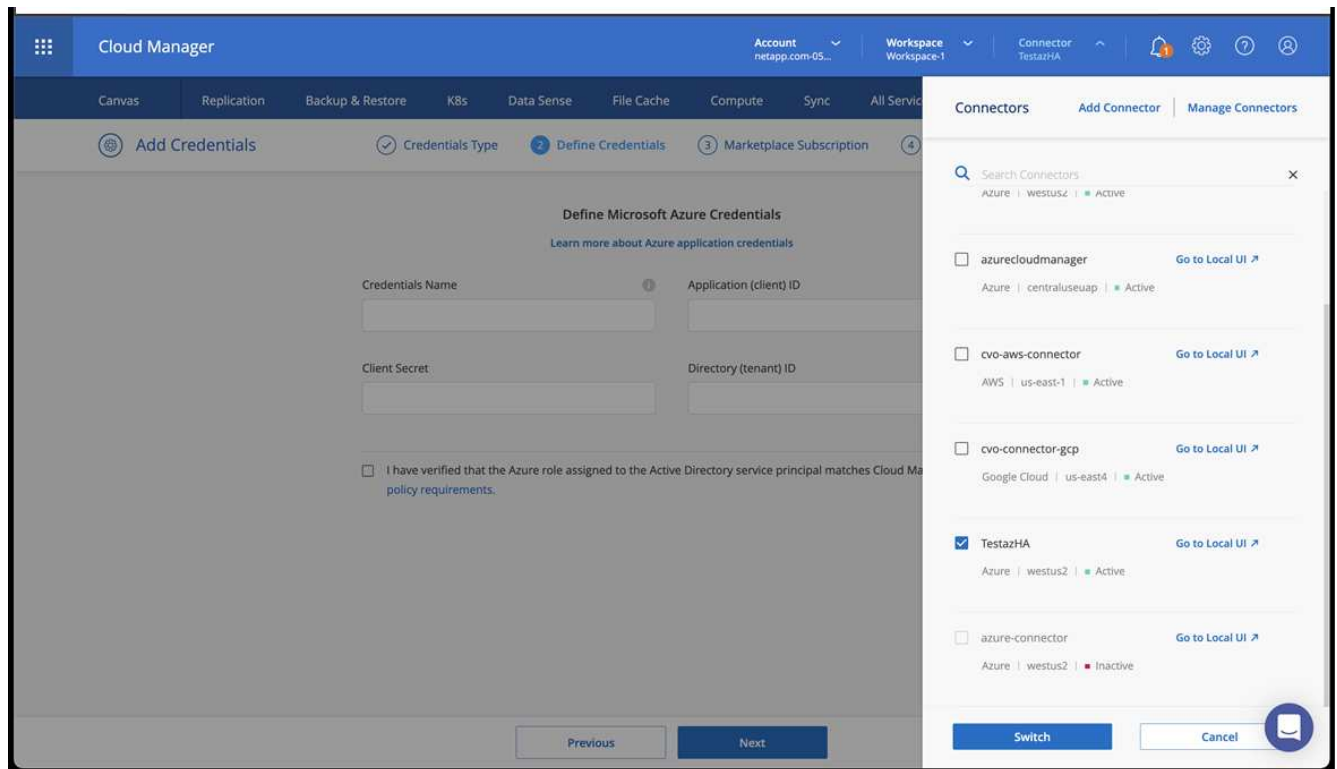
Zugriff auf das Azure Konto mit den erforderlichen IAM-Berechtigungen und -Rollen

Schritte

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

Siehe "[Erstellen eines Konnektors in Azure aus Cloud Manager](#)".

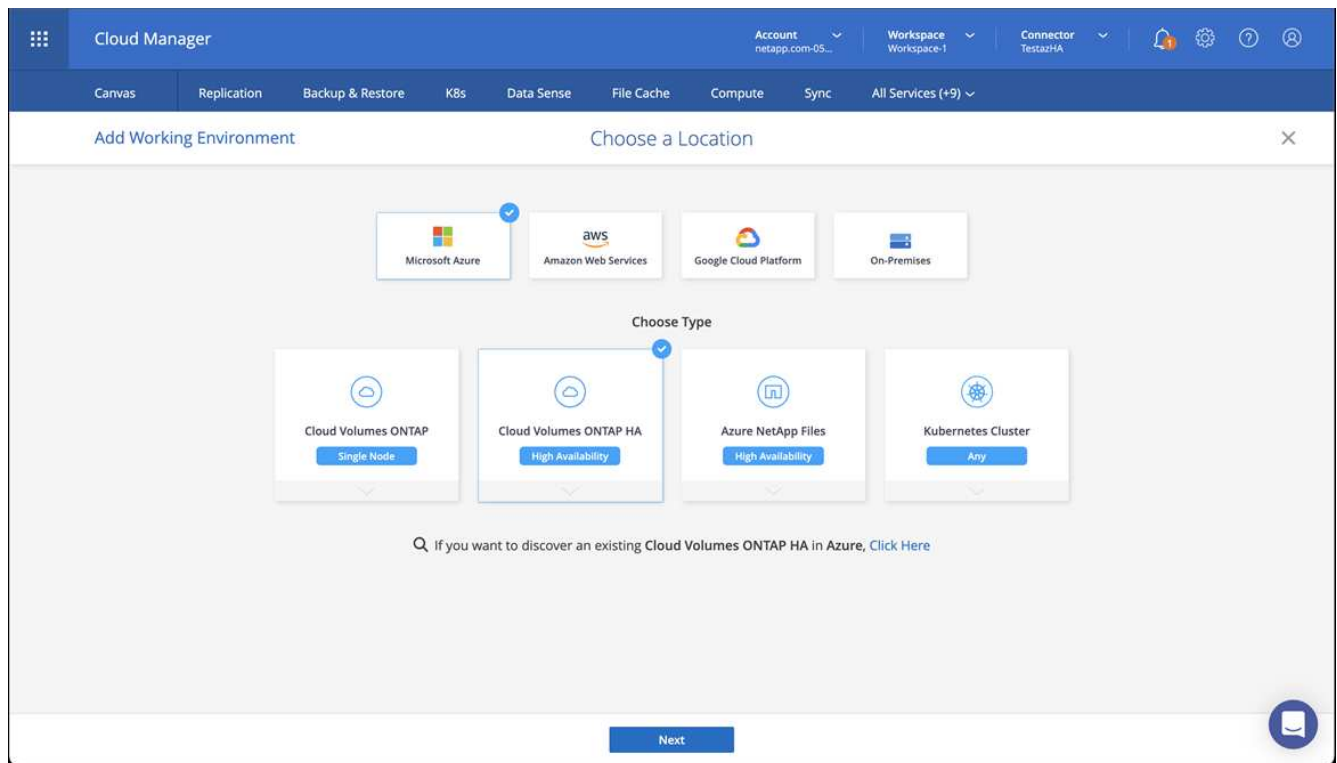
3. Stellen Sie sicher, dass der Anschluss läuft, und wechseln Sie zu diesem Anschluss.



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

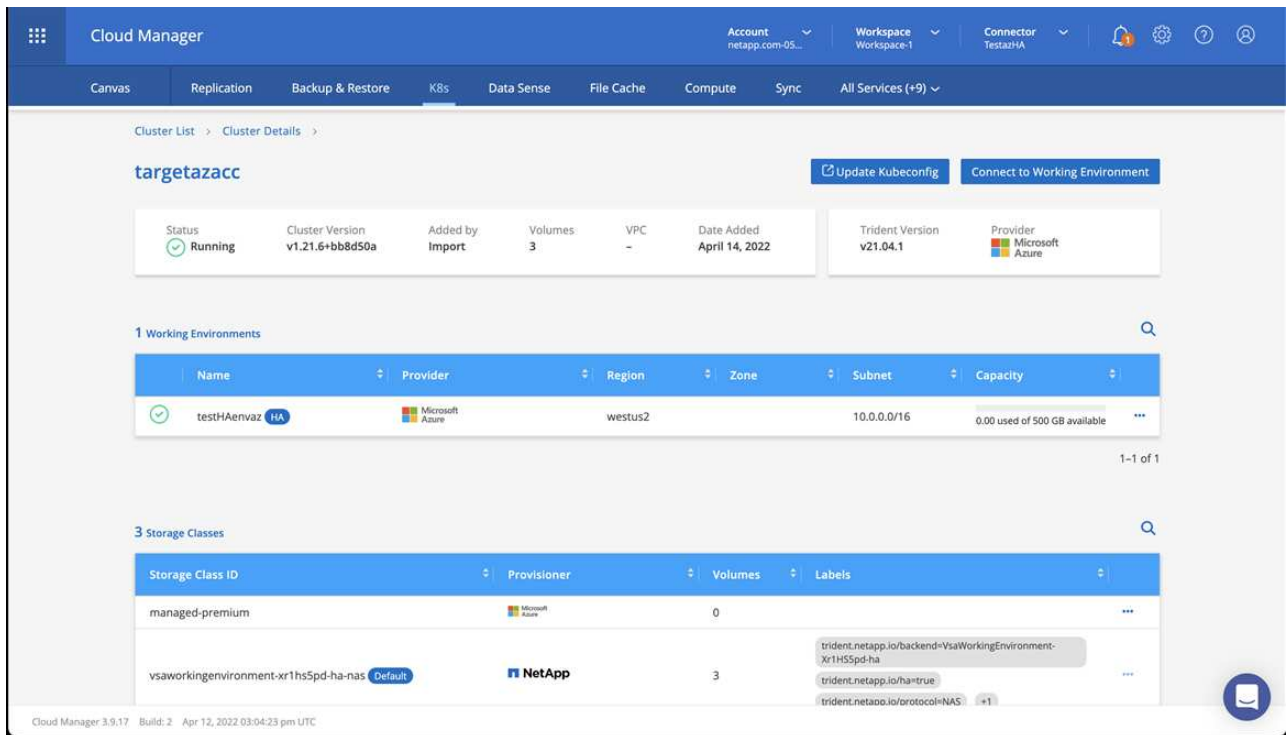
a. Ort: „Microsoft Azure“.

b. Typ: „Cloud Volumes ONTAP HA“.



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

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



b. Beachten Sie oben rechts die Trident-Version.

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

Damit wird Ihr Red hat OpenShift-Cluster importiert und eine Standard-speicherklasse zugewiesen. Sie wählen die Speicherklasse aus. Trident wird automatisch im Rahmen des Import- und Erkennungsvorgangs installiert.

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

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

Installation und Konfiguration des Astra Control Center

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

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

Einrichten des Astra Control Center

Astra Control Center unterstützt und überwacht ONTAP und Astra Data Store als Storage-Backend. Nach der Installation von Astra Control Center, melden Sie sich in der UI an und ändern Sie Ihr Passwort, Sie möchten eine Lizenz einrichten, Cluster hinzufügen, Speicher verwalten und Buckets hinzufügen.

Aufgaben

- [Fügen Sie eine Lizenz für Astra Control Center hinzu](#)
- [Cluster hinzufügen](#)

- [Fügen Sie ein Storage-Back-End hinzu](#)
- [Fügen Sie einen Bucket hinzu](#)

Fügen Sie eine Lizenz für Astra Control Center hinzu

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

Weitere Informationen zur Berechnung von Lizenzen finden Sie unter ["Lizenzierung"](#).



Informationen zum Aktualisieren einer vorhandenen Testversion oder Volllizenz finden Sie unter ["Aktualisieren einer vorhandenen Lizenz"](#).

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

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



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

Was Sie benötigen

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

Schritte

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

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



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

Cluster hinzufügen

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

Cluster hinzufügen, der Applikationen enthält, die von Astra Data Store bereitgestellte Volumes verwenden.



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



Wenn Astra Control einen Cluster verwaltet, wird die Standard-Storage-Klasse des Clusters überwacht. Wenn Sie die Speicherklasse mit ändern `kubectl` Befehle, Astra Control setzt die Änderung zurück. Verwenden Sie eine der folgenden Methoden, um die Standard-Storage-Klasse in einem von Astra Control gemanagten Cluster zu ändern:

- Verwenden Sie die Astra Control API `PUT /managedClusters` endpoint, und weisen Sie dem eine andere Standard-Speicherklasse zu `DefaultStorageClass` Parameter.
- Verwenden Sie die Web-UI von Astra Control, um eine andere Standard-Storage-Klasse zuzuweisen. Siehe [Ändern der Standard-Storage-Klasse](#).

Was Sie benötigen

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

Schritte

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



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



Add cluster

STEP 1/3: CREDENTIALS

CREDENTIALS

Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.

Follow [instructions](#) on how to create a dedicated admin-role kubeconfig.

Upload file

Paste from clipboard

Kubeconfig YAML file
No file selected



Credential name



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

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



Sie sollten sich für einen Trident-Storage-Kurs entscheiden, der von ONTAP Storage oder Astra Data Store unterstützt wird.

Add cluster
STEP 2/3: STORAGE

CONFIGURE STORAGE

Existing storage classes are discovered and verified as eligible for use with Astra. You can use your existing default, or choose to set a new default at this time.
Applications with persistent volumes on eligible storage classes are validated for use with Astra.

Default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
<input checked="" type="radio"/>	basic-csi	csi.trident.netapp.io	Delete		
<input type="radio"/>	thin	kubernetes.io/vsphere-volume	Delete		

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

Ergebnis

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



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

Fügen Sie ein Storage-Back-End hinzu

Sie können ein Storage-Backend hinzufügen, sodass Astra Control die Ressourcen managen kann. Sie können ein Storage-Back-End auf einem gemanagten Cluster implementieren oder ein vorhandenes Storage-Back-End verwenden.

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.

Was Sie für bestehende Implementierungen von Astra Data Store benötigen

- Sie haben Ihren Kubernetes-App-Cluster und das zugrunde liegende Computing-Cluster hinzugefügt.



Nachdem Sie Ihren Kubernetes App-Cluster für Astra Data Store hinzugefügt haben und er durch Astra Control gemanagt wird, erscheint der Cluster wie `unmanaged` In der Liste der entdeckten Back-Ends. Als Nächstes müssen Sie das Computing-Cluster hinzufügen, das Astra Data Store enthält und das Kubernetes App-Cluster untermauert. Dies können Sie über **Backends** in der UI tun. Wählen Sie das Menü Aktionen für den Cluster aus `Manage`, und **"Fügen Sie den Cluster hinzu"**. Nach dem Status des Clusters von `unmanaged` Änderungen am Namen des Kubernetes-Clusters können Sie mit dem Hinzufügen eines Backend fortfahren.

Was Sie für die neuen Astra Data Store Implementierungen benötigen

- Das ist schon **"Die Version des Installationspakets, das Sie bereitstellen möchten, hochgeladen haben"** Zu einem Ort, der für Astra Control zugänglich ist.
- Sie haben den Kubernetes-Cluster hinzugefügt, den Sie für die Implementierung verwenden möchten.
- Sie haben die hochgeladen **Astra Data Store-Lizenz** Für Ihre Implementierung an einen Standort, auf den Astra Control zugreifen kann.

Optionen

- [Implementieren von Storage-Ressourcen](#)
- [Verwenden Sie ein vorhandenes Storage-Back-End](#)

Implementieren von Storage-Ressourcen

Sie können einen neuen Astra Data Store implementieren und das zugehörige Storage-Backend verwalten.

Schritte

1. Navigieren Sie im Dashboard oder im Menü „Backend“:
 - Aus **Dashboard**: Wählen Sie in der Ressourcenübersicht einen Link aus dem Fensterbereich Speicherrückseite aus und wählen Sie im Bereich Back Ends **Add** aus.
 - Von **Backends**:
 - i. Wählen Sie im linken Navigationsbereich **Backend** aus.
 - ii. Wählen Sie **Hinzufügen**.
2. Wählen Sie auf der Registerkarte **Bereitstellen** die Option *** Astra Data Store* Deployment** aus.
3. Wählen Sie das zu implementierende Astra Data Store-Paket aus:
 - a. Geben Sie einen Namen für die Astra Data Store-Anwendung ein.
 - b. Wählen Sie die Version des Astra Data Stores, die Sie implementieren möchten.



Wenn Sie die Version, die Sie bereitstellen möchten, noch nicht hochgeladen haben, können Sie die Option **Paket hinzufügen** verwenden oder den Assistenten beenden und verwenden **"Paketmanagement"** Um das Installationspaket hochzuladen.

4. Wählen Sie eine Astra Data Store-Lizenz aus, die Sie bereits hochgeladen haben, oder laden Sie die **Lizenz hinzufügen**-Option ein, die Sie mit der Anwendung verwenden können.



Astra Data Store-Lizenzen mit vollständigen Berechtigungen sind mit Ihrem Kubernetes-Cluster verknüpft. Die zugehörigen Cluster sollten automatisch angezeigt werden. Wenn kein verwalteter Cluster vorhanden ist, können Sie die Option **Cluster hinzufügen** zur Astra Control-Verwaltung hinzufügen wählen. Für Astra Data Store-Lizenzen können Sie diese Verknüpfung auf der nächsten Seite des Assistenten definieren, wenn keine Verbindung zwischen Lizenz und Cluster hergestellt wurde.

5. Wenn Sie dem Astra Control Management noch kein Kubernetes-Cluster hinzugefügt haben, müssen Sie dies auf der Seite * Kubernetes Cluster* tun. Wählen Sie einen vorhandenen Cluster aus der Liste aus, oder wählen Sie **Hinzufügen des zugrunde liegenden Clusters** aus, um ein Cluster zum Astra Control Management hinzuzufügen.
6. Wählen Sie eine Vorlagengröße für den Kubernetes Cluster aus, die Ressourcen für Astra Data Store bereitstellen wird. Sie können eine der folgenden Optionen auswählen:
 - Wenn Sie sich entscheiden `Recommended Kubernetes worker node requirements`, Wählen Sie eine Vorlage von groß zu klein basierend auf, was Ihre Lizenz erlaubt.
 - Wenn Sie sich entscheiden `Custom Kubernetes worker node requirements`, Wählen Sie die Anzahl der Kerne und den gesamten Arbeitsspeicher aus, die Sie für jeden Cluster-Knoten benötigen. Sie können auch die zulässige Anzahl von Nodes im Cluster anzeigen, die die Auswahlkriterien für Kerne und Speicher erfüllen.



Wählen Sie bei der Auswahl einer Vorlage größere Nodes mit mehr Arbeitsspeicher und Kernen für größere Workloads oder eine größere Anzahl an Nodes für kleinere Workloads aus. Wählen Sie eine Vorlage basierend auf den von Ihrer Lizenz zulässt aus. Bei jeder empfohlenen Vorlagenoption wird die Anzahl der qualifizierten Nodes angegeben, die dem Vorlagenmuster für Arbeitsspeicher und Kerne sowie der Kapazität für jeden Node entsprechen.

7. Konfigurieren der Nodes:

- a. Fügen Sie eine Node-Bezeichnung hinzu, um den Pool der Worker-Nodes zu identifizieren, die diesen Astra Data Store-Cluster unterstützen.



Das Label muss jedem einzelnen Node im Cluster hinzugefügt werden, der vor Beginn der Implementierung oder der Implementierung von Astra Data Store genutzt wird.

- b. Konfigurieren Sie die Kapazität (gib) pro Node manuell, oder wählen Sie die maximal zulässige Node-Kapazität aus.
 - c. Konfigurieren Sie eine Höchstzahl der im Cluster zulässigen Nodes oder zulassen die maximale Anzahl der Nodes im Cluster.
8. (Nur Astra Data Store Vollizenzen) Geben Sie den Schlüssel des Etiketts ein, das Sie für Protection Domains verwenden möchten.



Erstellen Sie für jeden Node mindestens drei eindeutige Beschriftungen für den Schlüssel. Beispiel: Wenn Ihr Schlüssel lautet `astra.datastore.protection.domain`, Sie können die folgenden Etiketten erstellen:

```
astra.datastore.protection.domain=domain1,astra.datastore.protection
.domain=domain2, und astra.datastore.protection.domain=domain3.
```

9. Konfigurieren des Managementnetzwerks:

- a. Geben Sie eine Management-IP-Adresse für die interne Verwaltung von Astra Data Store ein, die sich im gleichen Subnetz wie die IP-Adressen der Worker-Nodes befindet.
 - b. Sie können dieselbe NIC sowohl für Management- als auch für Datennetzwerke verwenden oder sie separat konfigurieren.
 - c. Geben Sie den Daten-Netzwerk-IP-Adressenpool, die Subnetzmaske und das Gateway für den Storage-Zugriff ein.
10. Überprüfen Sie die Konfiguration und wählen Sie **Bereitstellen**, um mit der Installation zu beginnen.

Ergebnis

Nach erfolgreicher Installation erscheint das Backend in `available`. Geben Sie in der Back-Ends-Liste zusammen mit aktiven Performance-Informationen an.



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

Verwenden Sie ein vorhandenes Storage-Back-End

Sie können ein entdecktes ONTAP oder Astra Data Store Storage Back-End in das Astra Control Center Management integrieren.

Schritte

1. Navigieren Sie im Dashboard oder im Menü „Backend“:
 - Aus **Dashboard**: Wählen Sie in der Ressourcenübersicht einen Link aus dem Fensterbereich Speicherrückseite aus und wählen Sie im Bereich Back Ends **Add** aus.
 - Von **Backends**:
 - i. Wählen Sie im linken Navigationsbereich **Backend** aus.
 - ii. Wählen Sie **Verwalten** auf einem ermittelten Backend aus dem verwalteten Cluster oder wählen Sie **Hinzufügen**, um ein zusätzliches vorhandenes Backend zu verwalten.
2. Wählen Sie die Registerkarte **vorhandene** verwenden.
3. Je nach Backend-Typ:
 - **Astra Data Store**:
 - i. Wählen Sie **Astra Data Store**.
 - ii. Wählen Sie das verwaltete Compute-Cluster aus und wählen Sie **Next** aus.
 - iii. Bestätigen Sie die Back-End-Details und wählen Sie **Add Storage Backend**.
 - **ONTAP**:
 - i. Wählen Sie **ONTAP** und wählen Sie **Weiter**.
 - ii. Geben Sie die IP-Adresse und die Administrator-Anmeldedaten für das ONTAP-Cluster-Management ein.



Der Benutzer, dessen Anmeldeinformationen Sie hier eingeben, muss über den verfügbaren `ontapi` Aktivieren der Zugriffsmethode für die Anmeldung beim Benutzer in ONTAP System Manager auf dem ONTAP Cluster. Wenn Sie Vorhaben, SnapMirror Replizierung zu verwenden, aktivieren Sie die Zugriffsmethoden `ontapi` Und `http` Für den Benutzer auf beiden ONTAP Clustern. Siehe "[Benutzerkonten Verwalten](#)" Finden Sie weitere Informationen.

- iii. Wählen Sie **Bewertung**.
- iv. Bestätigen Sie die Back-End-Details und wählen Sie **Add Storage Backend**.

Ergebnis

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



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

Fügen Sie einen Bucket hinzu

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

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

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

Verwenden Sie einen der folgenden Bucket-Typen:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Allgemein S3



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

- Microsoft Azure



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

- Microsoft Azure

Anweisungen zum Hinzufügen von Buckets mithilfe der Astra Control API finden Sie unter "[Astra Automation und API-Informationen](#)".

Schritte

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



Wenn Sie einen Bucket hinzufügen, wählen Sie den richtigen Bucket-Provider 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.

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



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

- d. Geben Sie den Namen oder die IP-Adresse des S3-Endpunkts ein.
- e. Wenn dieser Bucket der Standard-Bucket für alle Backups sein soll, prüfen Sie den `Make this bucket the default bucket for this private cloud` Option.



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

- f. Mit **Hinzufügen** fortfahren [Anmeldeinformationen](#).

Fügen Sie S3-Zugriffsdaten hinzu

Fügen Sie Ihre Zugangsdaten für S3-Zugriff jederzeit hinzu.

Schritte

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

Ändern der Standard-Storage-Klasse

Sie können die Standard-Storage-Klasse für ein Cluster ändern.

Schritte

1. Wählen Sie in der Web-UI des Astra Control Center die Option **Cluster** aus.
2. Wählen Sie auf der Seite **Cluster** den Cluster aus, den Sie ändern möchten.
3. Wählen Sie die Registerkarte **Storage** aus.
4. Wählen Sie die Kategorie **Speicherklassen** aus.
5. Wählen Sie das Menü **Aktionen** für die Speicherklasse aus, die Sie als Standard festlegen möchten.
6. Wählen Sie **als Standard**.

Was kommt als Nächstes?

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

- ["Benutzer managen"](#)
- ["Starten Sie das Anwendungsmanagement"](#)
- ["Schützen von Applikationen"](#)

- ["Applikationen klonen"](#)
- ["Benachrichtigungen verwalten"](#)
- ["Verbinden Sie sich mit Cloud Insights"](#)
- ["Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu"](#)

Weitere Informationen

- ["Verwenden Sie die Astra Control API"](#)
- ["Bekannte Probleme"](#)

Voraussetzungen für das Hinzufügen eines Clusters

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

Was benötigen Sie vor dem Hinzufügen eines Clusters

Vergewissern Sie sich, dass Ihr Cluster die in aufgeführten Anforderungen erfüllt ["Anforderungen für Applikationscluster"](#).



Wenn Sie planen, als verwaltete Computing-Ressource einen zweiten OpenShift 4.6, 4.7 oder 4.8 hinzuzufügen, sollten Sie sicherstellen, dass die Astra Trident Volume Snapshot-Funktion aktiviert ist. Sehen Sie den offiziellen Astra Trident an ["Anweisungen"](#) Um Volume Snapshots mit Astra Trident zu aktivieren und zu testen.

- Astra Trident StorageClasses ist mit einem konfiguriert ["Unterstütztes Storage-Backend"](#) (Erforderlich für jeden Cluster-Typ)
- Der Superuser und die Benutzer-ID sind auf dem ONTAP-System eingerichtet, um Apps mit Astra Control Center zu sichern und wiederherzustellen. Führen Sie den folgenden Befehl in der ONTAP-Befehlszeile aus:

```
export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sysm --anon 65534
```
- Astra Trident `volumesnapshotclass` Objekt, das von einem Administrator definiert wurde. Astra Trident ist der Anfang ["Anweisungen"](#) Um Volume Snapshots mit Astra Trident zu aktivieren und zu testen.
- Stellen Sie sicher, dass nur eine einzelne Standard-Storage-Klasse für Ihr Kubernetes-Cluster definiert ist.

Führen Sie Eignungsprüfungen durch

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

Schritte

1. Überprüfen Sie die Trident Version.

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

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

```
NAME      VERSION
trident   21.04.0
```

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

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



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

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

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

Erstellen Sie ein „admin-Role“-kubeconfig

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

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

Schritte

1. Erstellen Sie ein Service-Konto wie folgt:

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

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

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

+

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

a. Wenden Sie das Servicekonto an:

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

2. (Optional) Wenn Ihr Cluster eine restriktive Pod-Sicherheitsrichtlinie verwendet, die die Erstellung privilegierter Pods nicht zulässt oder Vorgänge innerhalb der Pod-Container als Root-Benutzer ausgeführt werden können, erstellen Sie eine benutzerdefinierte Pod-Sicherheitsrichtlinie für den Cluster, durch die Astra Control Pods erstellen und managen kann. Anweisungen hierzu finden Sie unter "[Erstellen einer benutzerdefinierten POD-Sicherheitsrichtlinie](#)".

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

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

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

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

+

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

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

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

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 in einen aussagekräftigen Namen für Ihren Cluster. Schützen Sie die Cluster-Anmeldedaten.

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

Was kommt als Nächstes?

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

Weitere Informationen

- "[Trident Dokumentation](#)"
- "[Verwenden Sie die Astra Control API](#)"

Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

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

Was Sie benötigen

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

Entfernen Sie das selbstsignierte Zertifikat

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 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>
```

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

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

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

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

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

5. Erstellen Sie das Zertifikat mit dem folgenden Befehl:

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

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

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

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

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

```

kubect1 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.

Erstellen einer benutzerdefinierten POD-Sicherheitsrichtlinie

Astra Control muss Kubernetes Pods auf den gemanagten Clustern erstellen und managen. Wenn Ihr Cluster eine restriktive Pod-Sicherheitsrichtlinie verwendet, die die Erstellung privilegierter Pods nicht zulässt oder Vorgänge innerhalb der Pod-Container nicht als Root-Benutzer ausgeführt werden können, müssen Sie eine weniger restriktive POD-Sicherheitsrichtlinie erstellen, damit Astra Control diese Pods erstellen und verwalten kann.

Schritte

1. Erstellen Sie eine Pod-Sicherheitsrichtlinie für den Cluster, die weniger restriktiv ist als der Standard, und speichern Sie sie in einer Datei. Beispiel:

```

apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'

```

2. Erstellen Sie eine neue Rolle für die POD-Sicherheitsrichtlinie.

```

kubectl-admin create role psp:astracontrol \
  --verb=use \
  --resource=podsecuritypolicy \
  --resource-name=astracontrol

```

3. Binden Sie die neue Rolle an das Dienstkonto.

```

kubectl-admin create rolebinding default:psp:astracontrol \
  --role=psp:astracontrol \
  --serviceaccount=astracontrol-service-account:default

```

Häufig gestellte Fragen zum Astra Control Center

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

Überblick

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

Zugang zum Astra Control Center

Was ist die Astra Control URL?

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

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

Lizenzierung

Ich verwende die Evaluierungslizenz. Wie ändere ich die Volllizenz?

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

Schritte

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

Ich verwende die Evaluierungslizenz. Kann ich trotzdem Apps verwalten?

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

Kubernetes Cluster werden registriert

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

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

Wie entnehme ich einen Cluster richtig?

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

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

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



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

Wird NetApp Trident bei Unmanagement automatisch von einem Cluster deinstalliert? Wenn Sie das Management eines Clusters aus dem Astra Control Center aufheben, wird Trident nicht automatisch aus dem Cluster deinstalliert. Um Trident zu deinstallieren, müssen Sie es benötigen "[Befolgen Sie die folgenden Schritte in der Trident-Dokumentation](#)".

Management von Applikationen

Kann Astra Control eine Anwendung bereitstellen?

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

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

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

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

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

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

Beeinträchtigen ungesunde Pods das App-Management? Wenn eine verwaltete App Pods in einem ungesunden Zustand hat, kann Astra Control keine neuen Backups und Klone erstellen.

Datenmanagement-Vorgänge

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

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

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

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

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

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

Copyright-Informationen

Copyright © 2024 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.