Fügen Sie einen Cluster hinzu

Astra Control Service

NetApp
April 24, 2024

Inhalt

Fügen Sie dem Astra Control Service einen Cluster hinzu 1
Astra Connector installieren, um Cluster zu managen 1
Fügen Sie ein vom Anbieter verwaltetes Cluster hinzu 7
Hinzufügen eines selbstverwalteten Clusters 18

Fügen Sie dem Astra Control Service einen Cluster hinzu

Nach der Einrichtung der Umgebung erstellen Sie sofort einen Kubernetes Cluster und fügen ihn dann dem Astra Control Service hinzu. Auf diese Weise können Sie Astra Control Service zum Schutz Ihrer Anwendungen auf dem Cluster verwenden.

Je nach dem Cluster-Typ, den Sie zum Astra Control Service hinzufügen müssen, müssen Sie den Cluster mit verschiedenen Schritten hinzufügen.

- "Fügen Sie Astra Control Service einen über einen öffentlichen Provider gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine öffentliche IP-Adresse hat und von einem Cloud-Provider verwaltet wird. Sie benötigen das Service Principal-Konto, das Service-Konto oder das Benutzerkonto für den Cloud-Provider.
- "Fügen Sie dem Astra Control Service einen über privaten Provider gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine private IP-Adresse hat und von einem Cloud-Provider verwaltet wird. Sie benötigen das Service Principal-Konto, das Service-Konto oder das Benutzerkonto für den Cloud-Provider.
- "Fügen Sie Astra Control Service einen öffentlichen, selbst gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine öffentliche IP-Adresse hat und von Ihrer Organisation verwaltet wird. Sie müssen eine kubeconfig-Datei für den Cluster erstellen, den Sie hinzufügen möchten.
- "Fügen Sie Astra Control Service einen privaten, selbst gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine private IP-Adresse hat und von Ihrer Organisation verwaltet wird. Sie müssen eine kubeconfig-Datei für den Cluster erstellen, den Sie hinzufügen möchten.

Astra Connector installieren, um Cluster zu managen

Astra Connector ist eine Software, die sich auf Ihren gemanagten Clustern befindet und die Kommunikation zwischen dem gemanagten Cluster und Astra Control erleichtert. Für Cluster, die mit Astra Control Service verwaltet werden, stehen zwei Versionen von Astra Connector zur Verfügung:

- **Frühere Version des Astra Connectors**: "Installieren Sie die vorherige Version von Astra Connector" In Ihrem Cluster, wenn Sie den Cluster mit nicht-Kubernetes-nativen Workflows managen möchten.
- [Tech Preview] **Declarative Kubernetes Astra Connector**: "Installieren Sie Astra Connector für Cluster, die mit deklarativen Kubernetes-Workflows gemanagt werden" Wenn Sie den Cluster mit deklarativen Kubernetes-Workflows managen möchten, befinden Sie sich in Ihrem Cluster. Nachdem Sie den Astra Connector auf Ihrem Cluster installiert haben, wird der Cluster automatisch zu Astra Control hinzugefügt.

> Der deklarative Kubernetes Astra Connector ist nur im Rahmen des Astra Control Early Adopter Program (EAP) verfügbar. Informationen zum Beitritt zum EAP erhalten Sie von Ihrem NetApp Ansprechpartner.

Installieren Sie die vorherige Version von Astra Connector

Astra Control Service verwendet die vorherige Version von Astra Connector, um die Kommunikation zwischen Astra Control Service und privaten Clustern zu ermöglichen,

die über nicht-Kubernetes-native Workflows gemanagt werden. Sie müssen Astra Connector auf privaten Clustern installieren, die Sie mit nicht-Kubernetes-nativen Workflows managen möchten.

Die vorherige Version von Astra Connector unterstützt die folgenden Typen von privaten Clustern, die mit nicht-Kubernetes-nativen Workflows gemanagt werden:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes-Service (AKS)
- Google Kubernetes Engine (GKE)
- Red hat OpenShift Service auf AWS (ROSA)
- ROSA mit AWS PrivateLink
- Red hat OpenShift-Container-Plattform vor Ort

Über diese Aufgabe

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

Bevor Sie beginnen

- Sie benötigen Zugriff auf den privaten Cluster, den Sie mit Astra Control Service managen möchten.
- Sie benötigen Kubernetes-Administratorberechtigungen, um den Astra Connector Operator auf dem Cluster zu installieren.

Schritte

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

   ```
   kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/23.07.0-202310251519/astraconnector_operator.yaml
   ```

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

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

3. Holen Sie sich ein API-Token von Astra Control. Siehe "Dokumentation von Astra Automation" Weitere Anweisungen.
4. astra-Connector-Namespace erstellen:

   ```
   kubectl create ns astra-connector
   ```

5. Erstellen Sie die Astra Connector CR-Datei und benennen Sie sie `astra-connector-cr.yaml`. Aktualisieren Sie die Werte in Klammern <>, um sie an die Astra Control-Umgebung und die Cluster-Konfiguration anzupassen:
   - **<ASTRA_CONTROL_SERVICE_URL>**: Die Web UI URL des Astra Control Service. Beispiel:

     ```
     https://astra.netapp.io
     ```

   - **<ASTRA_CONTROL_SERVICE_API_TOKEN>**: Das Astra Control API Token, das Sie im vorherigen Schritt erhalten haben.
   - **<PRIVATE_AKS_CLUSTER_NAME>**: (Nur AKS-Cluster) - der Cluster-Name des privaten Azure Kubernetes Service Clusters. Heben Sie die Kommentareingabe auf und füllen Sie diese Zeile nur dann aus, wenn Sie einen privaten AKS-Cluster hinzufügen.
   - **<ASTRA_CONTROL_ACCOUNT_ID>**: Erhalten von der Astra Control Web-Benutzeroberfläche. Wählen Sie das Symbol oben rechts auf der Seite aus und wählen Sie **API Access** aus.

     ```yaml
     apiVersion: netapp.astraconnector.com/v1
     kind: AstraConnector
     metadata:
       name: astra-connector
       namespace: astra-connector
     spec:
       natssync-client:
         cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
       imageRegistry:
         name: theotw
         secret: ""
       astra:
         token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
         #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
         accountId: <ASTRA_CONTROL_ACCOUNT_ID>
         acceptEULA: yes
     ```

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

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

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

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

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

```
kubectl get astraconnector -n astra-connector
```

Sie sollten eine Ausgabe wie die folgende sehen:

```
NAME                REGISTERED   ASTRACONNECTORID
STATUS
astra-connector     true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

Notieren Sie sich die ASTRACONNECTORID, die Sie benötigen, wenn Sie den Cluster zu Astra Control hinzufügen.

Was kommt als Nächstes?

Nachdem Sie jetzt Astra Connector installiert haben, können Sie jetzt Ihrem privaten Cluster den Astra Control Service hinzufügen.

- "Fügen Sie dem Astra Control Service einen über privaten Provider gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine private IP-Adresse hat und von einem Cloud-Provider verwaltet wird. Sie benötigen das Service Principal-Konto, das Service-Konto oder das Benutzerkonto für den Cloud-Provider.
- "Fügen Sie Astra Control Service einen privaten, selbst gemanagten Cluster hinzu": Verwenden Sie diese Schritte, um einen Cluster hinzuzufügen, der eine private IP-Adresse hat und von Ihrer Organisation verwaltet wird. Sie müssen eine kubeconfig-Datei für den Cluster erstellen, den Sie hinzufügen möchten.

Finden Sie weitere Informationen

- "Fügen Sie einen Cluster hinzu"

(Tech Preview) Installieren Sie den deklarativen Kubernetes Astra Connector

Cluster, die über deklarative Kubernetes-Workflows gemanagt werden, ermöglichen über Astra Connector die Kommunikation zwischen dem gemanagten Cluster und Astra Control. Sie müssen Astra Connector auf allen Clustern installieren, die Sie mit deklarativen Kubernetes-Workflows managen werden.

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

Über diese Aufgabe

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

Bevor Sie beginnen

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

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

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

Schritte

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Sie sollten eine Ausgabe wie die folgende sehen:

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

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

Fügen Sie ein vom Anbieter verwaltetes Cluster hinzu

Fügen Sie Astra Control Service einen über einen öffentlichen Provider gemanagten Cluster hinzu

Nachdem Sie Ihre Cloud-Umgebung eingerichtet haben, sind Sie bereit, ein Kubernetes-Cluster zu erstellen und dieses dann zu Astra Control Service hinzuzufügen.

- Erstellen eines Kubernetes-Clusters
- Fügen Sie das Cluster zu Astra Control Service hinzu
- Ändern der Standard-Storage-Klasse

Erstellen eines Kubernetes-Clusters

Wenn Sie noch keinen Cluster haben, können Sie ein Cluster erstellen, das erfüllt "Astra Control Service-Anforderungen für Amazon Elastic Kubernetes Service (EKS)". Wenn Sie noch keinen Cluster haben, können Sie ein Cluster erstellen, das erfüllt "Astra Control Service-Anforderungen für Google Kubernetes Engine (GKE)". Wenn Sie noch keinen Cluster haben, können Sie ein Cluster erstellen, das erfüllt "Astra Control Service-Anforderungen für Azure Kubernetes Service (AKS) mit Azure NetApp Files" Oder "Astra Control Service-Anforderungen für Azure Kubernetes Service (AKS) mit von Azure gemanagten Festplatten".

> Astra Control Service unterstützt AKS-Cluster, die Azure Active Directory (Azure AD) zur Authentifizierung und Identitätsverwaltung nutzen. Wenn Sie das Cluster erstellen, befolgen Sie die Anweisungen im "Offizielle Dokumentation" Um den Cluster mit Azure AD zu konfigurieren. Stellen Sie sicher, dass Ihre Cluster die Anforderungen für die AKS-verwaltete Azure AD-Integration erfüllen.

Fügen Sie das Cluster zu Astra Control Service hinzu

Nachdem Sie sich beim Astra Control Service angemeldet haben, beginnen Sie zunächst mit dem Verwalten Ihrer Cluster. Bevor Sie Astra Control Service ein Cluster hinzufügen, müssen Sie bestimmte Aufgaben ausführen und sicherstellen, dass das Cluster bestimmte Anforderungen erfüllt.

Beachten Sie beim Management von Azure Kubernetes Service und Google Kubernetes Engine-Clustern, dass für die Installation von Astra Control und das Lifecycle Management zwei Optionen zur Verfügung stehen:

- Mit Astra Control Service können Sie den Lebenszyklus von Astra Control Provisioner automatisch managen. Vergewissern Sie sich dazu, dass Astra Trident nicht installiert ist und Astra Control Provisioner nicht auf dem Cluster aktiviert ist, den Sie mit Astra Control Service managen möchten. In diesem Fall aktiviert Astra Control Service automatisch die Astra Control-Bereitstellung, wenn Sie mit dem Cluster-Management beginnen. Upgrades für die Astra Control-Bereitstellung werden automatisch durchgeführt.
- Sie können den Lebenszyklus der Astra Control Provisionierung selbst managen. Aktivieren Sie hierfür die Astra Control-Provisionierung im Cluster, bevor Sie das Cluster mit Astra Control Service verwalten. In diesem Fall erkennt Astra Control Service, dass die Provisionierung von Astra Control bereits aktiviert ist. Es wird weder neu installiert noch Astra Control Provisioner-Upgrades gemanagt. Siehe "Astra Control Provisioner Aktivieren" Für die Schritte aktivieren Sie die Astra Control-Provisionierung.

Wenn Sie Amazon Web Services Cluster mit Astra Control Service managen, müssen Sie bei Bedarf Storage-Back-Ends, die nur mit dem Astra Control Provisioner verwendet werden können, die Astra Control Service manuell im Cluster aktivieren, bevor Sie die Bereitstellung mit Astra Control Service managen. Siehe "Astra Control Provisioner Aktivieren" Enthält die Schritte zum Aktivieren der Astra Control-Bereitstellung.

Bevor Sie beginnen

Amazon Web Services

- Sie sollten die JSON-Datei mit den Anmeldedaten des IAM-Benutzers haben, der das Cluster erstellt hat. "Erfahren Sie, wie ein IAM-Benutzer erstellt wird".
- Astra Control Provisioner ist für Amazon FSX for NetApp ONTAP erforderlich. Wenn Sie Amazon FSX for NetApp ONTAP als Storage-Backend für Ihr EKS-Cluster verwenden möchten, finden Sie in den Informationen zur Astra Control-Bereitstellung im "EKS-Clusteranforderungen".
- (Optional) Wenn Sie angeben müssen `kubectl` Befehlszugriff für ein Cluster auf andere IAM-Benutzer, die nicht der Ersteller des Clusters sind, finden Sie in den Anweisungen unter "Wie erhalte ich Zugriff auf andere IAM-Benutzer und Rollen nach der Cluster-Erstellung in Amazon EKS?".
- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Nutzung mit Amazon Web Services konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Microsoft Azure

- Sie sollten beim Erstellen des Service-Principal die JSON-Datei haben, die die Ausgabe aus der Azure CLI enthält. "Erfahren Sie, wie Sie einen Service-Principal einrichten".

  Außerdem benötigen Sie Ihre Azure Abonnement-ID, wenn Sie sie nicht zur JSON-Datei hinzugefügt haben.

- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Zusammenarbeit mit Microsoft Azure konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Google Cloud

- Sie sollten die Servicekontoschlüsseldatei für ein Servicekonto haben, das über die erforderlichen Berechtigungen verfügt. "Erfahren Sie, wie Sie ein Service-Konto einrichten".
- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Zusammenarbeit mit Google Cloud konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Schritte

1. (Optional) Wenn Sie einen Amazon EKS Cluster hinzufügen oder die Installation und Upgrades von Astra Control Provisioner selbst managen möchten, aktivieren Sie die Astra Control Provisioner-Funktion im Cluster. Siehe "Astra Control Provisioner Aktivieren" Für Enablement-Schritte.
2. Öffnen Sie die Web-UI des Astra Control Service in einem Browser.
3. Wählen Sie im Dashboard **Kubernetes Cluster managen** aus.

   Befolgen Sie die Aufforderungen zum Hinzufügen des Clusters.

4. **Provider**: Wählen Sie Ihren Cloud-Provider aus und geben Sie dann entweder die erforderlichen Anmeldedaten für die Erstellung einer neuen Cloud-Instanz an, oder wählen Sie eine vorhandene Cloud-Instanz aus.
5. **Amazon Web Services**: Geben Sie Details über Ihr Amazon Web Services IAM-Benutzerkonto an, indem Sie eine JSON-Datei hochladen oder den Inhalt dieser JSON-Datei aus Ihrer Zwischenablage einfügen.

   Die JSON-Datei sollte die Anmeldeinformationen des IAM-Benutzers enthalten, der das Cluster erstellt hat.

6. **Microsoft Azure**: Geben Sie Details zu Ihrem Azure Service Principal an, indem Sie eine JSON-Datei hochladen oder den Inhalt dieser JSON-Datei aus Ihrer Zwischenablage einfügen.

   Die JSON-Datei sollte beim Erstellen des Service-Principal die Ausgabe aus der Azure CLI enthalten. Sie können auch Ihre Abonnement-ID angeben, damit sie automatisch in den Astra aufgenommen wird. Andernfalls müssen Sie die ID manuell eingeben, nachdem Sie den JSON bereitgestellt haben.

7. **Google Cloud Platform**: Stellen Sie die Service-Konto-Schlüsseldatei entweder durch das Hochladen der Datei oder durch Einfügen der Inhalte aus Ihrer Zwischenablage bereit.

   Astra Control Service nutzt das Service-Konto, um Cluster zu erkennen, die in der Google Kubernetes Engine ausgeführt werden.

8. **Andere**: Diese Registerkarte ist nur für die Verwendung mit selbst verwalteten Clustern vorgesehen.
   a. **Cloud-Instanzname**: Geben Sie einen Namen für die neue Cloud-Instanz an, die beim Hinzufügen dieses Clusters erstellt wird. Weitere Informationen zu "Cloud-Instanzen".
   b. Wählen Sie **Weiter**.

      Astra Control Service zeigt eine Liste von Clustern an, aus denen Sie auswählen können.

   c. **Cluster**: Wählen Sie einen Cluster aus der Liste aus, der zu Astra Control Service hinzugefügt werden soll.

      > Wenn Sie aus der Liste der Cluster auswählen, achten Sie auf die Spalte **Eligiblity**. Wenn ein Cluster „nicht berechtigt“ oder „teilweise berechtigt“ ist, bewegen Sie den Mauszeiger über den Status, um zu ermitteln, ob ein Problem im Cluster vorliegt. Beispielsweise kann sie erkennen, dass für das Cluster kein Worker Node vorhanden ist.

   d. Wählen Sie **Weiter**.
   e. (Optional) **Speicher**: Wählen Sie optional die Storage-Klasse aus, die Kubernetes-Anwendungen, die auf diesem Cluster bereitgestellt werden sollen, standardmäßig verwenden sollen.
9. Um eine neue Standard-Storage-Klasse für den Cluster auszuwählen, aktivieren Sie das Kontrollkästchen **Neue Standard-Storage-Klasse zuweisen**.
10. Wählen Sie eine neue Standard-Storage-Klasse aus der Liste aus.

Jeder Storage-Service eines Cloud-Providers enthält die folgenden Informationen zu Preis, Performance und Ausfallsicherheit:

- Cloud Volumes Service für Google Cloud: Informationen zu Preis, Performance und Ausfallsicherheit
- Google Persistent Disk: Keine Informationen über Preis, Performance oder Ausfallsicherheit verfügbar
- Azure NetApp Files: Informationen zu Performance und Ausfallsicherheit
- Azure Managed Disks: Es sind weder Preis-, Performance- oder Resilience-Informationen verfügbar
- Amazon Elastic Block Store: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar
- Amazon FSX für NetApp ONTAP: Keine Informationen zu Preis, Performance und Ausfallsicherheit verfügbar
- NetApp Cloud Volumes ONTAP: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar

Jede Storage-Klasse kann einen der folgenden Services nutzen:

- "Cloud Volumes Service für Google Cloud"
- "Google Persistent Disk"
  - "Azure NetApp Dateien"
  - "Von Azure gemanagte Festplatten"
  - "Amazon Elastic Block Store"
  - "Amazon FSX für NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

  Weitere Informationen zu "Storage-Klassen für Amazon Web Services Cluster". Weitere Informationen zu "Speicherklassen für AKS-Cluster". Weitere Informationen zu "Speicherklassen für GKE-Cluster".

  a. Wählen Sie **Weiter**.
  b. **Überprüfen und genehmigen**: Überprüfen Sie die Konfigurationsdetails.
  c. Wählen Sie **Add**, um den Cluster zu Astra Control Service hinzuzufügen.

Ergebnis

Wenn dies der erste Cluster ist, den Sie für diesen Cloud-Provider hinzugefügt haben, erstellt Astra Control Service einen Objektspeicher für den Cloud-Provider für Backups von Anwendungen, die auf geeigneten Clustern ausgeführt werden. (Wenn Sie nachfolgende Cluster für diesen Cloud-Provider hinzufügen, werden keine weiteren Objektspeicher erstellt.) Wenn Sie eine Standard-Storage-Klasse angegeben haben, setzt Astra Control Service die von Ihnen angegebene Standard-Storage-Klasse ein. Für Cluster, die in Amazon Web Services oder Google Cloud Platform gemanagt werden, erstellt Astra Control Service auch ein Administratorkonto auf dem Cluster. Diese Vorgänge können mehrere Minuten dauern.

Ändern der Standard-Storage-Klasse

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

Ändern Sie die Standard-Storage-Klasse mit Astra Control

Sie können die Standard-Storage-Klasse für ein Cluster aus Astra Control ändern. Wenn Ihr Cluster einen zuvor installierten Speicher-Backend-Service verwendet, können Sie diese Methode möglicherweise nicht verwenden, um die Standard-Speicherklasse zu ändern (die Aktion **default** ist nicht wählbar). In diesem Fall können Sie Ändern Sie die Standard-Storage-Klasse über die Befehlszeile.

Schritte

1. Wählen Sie in der Astra Control Service-UI **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**.

Ändern Sie die Standard-Storage-Klasse über die Befehlszeile

Sie können die Standard-Storage-Klasse für ein Cluster mit Kubernetes-Befehlen ändern. Diese Methode funktioniert unabhängig von der Konfiguration Ihres Clusters.

Schritte

1. Melden Sie sich bei Ihrem Kubernetes Cluster an.
2. Listen Sie die Storage-Klassen in Ihrem Cluster auf:

```
kubectl get storageclass
```

3. Entfernen Sie die Standardbezeichnung aus der Standardspeicherklasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Markieren Sie standardmäßig eine andere Storage-Klasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Bestätigen Sie die neue Standard-Speicherklasse:

```
kubectl get storageclass
```

Fügen Sie dem Astra Control Service einen über privaten Provider gemanagten Cluster hinzu

Mit Astra Control Service können Sie private GKE-Cluster (Google Kubernetes Engine) managen. In diesen Anweisungen wird davon ausgegangen, dass Sie bereits einen privaten AKS- oder OpenShift-Cluster erstellt und eine sichere Methode für den Remote-Zugriff darauf vorbereitet haben. Weitere Informationen zum Erstellen und Zugreifen auf private AKS- oder OpenShift-Cluster finden Sie in der folgenden Dokumentation:

- "Azure-Dokumentation für private AKS-Cluster"
- "Azure-Dokumentation für private OpenShift-Cluster"

Mit Astra Control Service können Sie private Azure Kubernetes Service (AKS)-Cluster sowie private Red hat OpenShift-Cluster in AKS managen. In diesen Anweisungen wird davon ausgegangen, dass Sie bereits einen privaten AKS- oder OpenShift-Cluster erstellt und eine sichere Methode für den Remote-Zugriff darauf vorbereitet haben. Weitere Informationen zum Erstellen und Zugreifen auf private AKS- oder OpenShift-Cluster finden Sie in der folgenden Dokumentation:

- "Azure-Dokumentation für private AKS-Cluster"
- "Azure-Dokumentation für private OpenShift-Cluster"

Mit Astra Control Service können Sie private EKS-Cluster (Amazon Elastic Kubernetes Service) managen. In diesen Anweisungen wird davon ausgegangen, dass Sie bereits einen privaten EKS-Cluster erstellt und eine sichere Methode für den Remote-Zugriff darauf vorbereitet haben. Weitere Informationen zum Erstellen und Zugreifen auf private EKS-Cluster finden Sie im "Amazon EKS-Dokumentation".

Führen Sie die folgenden Aufgaben aus, um Ihren privaten Cluster zum Astra Control Service hinzuzufügen:

1. Astra Connector Installieren
2. Einrichtung von persistentem Storage
3. Fügen Sie den über den privaten Provider gemanagten Cluster zu Astra Control Service hinzu

Astra Connector Installieren

Bevor Sie einen privaten Cluster hinzufügen, müssen Sie Astra Connector im Cluster installieren, damit Astra Control damit kommunizieren kann. Siehe "Installieren Sie die vorherige Version von Astra Connector für private Cluster, die mit nicht-Kubernetes-nativen Workflows gemanagt werden" Weitere Anweisungen.

Einrichtung von persistentem Storage

Konfigurieren Sie persistenten Storage für das Cluster. In der Dokumentation „erste Schritte“ finden Sie weitere Informationen zum Konfigurieren von persistentem Storage:

- "Microsoft Azure mit Azure NetApp Files einrichten"
- "Richten Sie Microsoft Azure mit von Azure gemanagten Festplatten ein"
- "Einrichten von Amazon Web Services"
- "Google Cloud einrichten"

Fügen Sie den über den privaten Provider gemanagten Cluster zu Astra Control Service hinzu

Sie können den privaten Cluster jetzt dem Astra Control Service hinzufügen.

Beachten Sie beim Management von Azure Kubernetes Service und Google Kubernetes Engine-Clustern, dass für die Installation von Astra Control und das Lifecycle Management zwei Optionen zur Verfügung stehen:

- Mit Astra Control Service können Sie den Lebenszyklus von Astra Control Provisioner automatisch managen. Vergewissern Sie sich dazu, dass Astra Trident nicht installiert ist und Astra Control Provisioner nicht auf dem Cluster aktiviert ist, den Sie mit Astra Control Service managen möchten. In diesem Fall aktiviert Astra Control Service automatisch die Astra Control-Bereitstellung, wenn Sie mit dem Cluster-Management beginnen. Upgrades für die Astra Control-Bereitstellung werden automatisch durchgeführt.
- Sie können den Lebenszyklus der Astra Control Provisionierung selbst managen. Aktivieren Sie hierfür die Astra Control-Provisionierung im Cluster, bevor Sie das Cluster mit Astra Control Service verwalten. In diesem Fall erkennt Astra Control Service, dass die Provisionierung von Astra Control bereits aktiviert ist. Es wird weder neu installiert noch Astra Control Provisioner-Upgrades gemanagt. Siehe "Astra Control Provisioner Aktivieren" Für die Schritte aktivieren Sie die Astra Control-Provisionierung.

Wenn Sie Amazon Web Services Cluster mit Astra Control Service managen, müssen Sie bei Bedarf Storage-Back-Ends, die nur mit dem Astra Control Provisioner verwendet werden können, die Astra Control Service manuell im Cluster aktivieren, bevor Sie die Bereitstellung mit Astra Control Service managen. Siehe "Astra Control Provisioner Aktivieren" Enthält die Schritte zum Aktivieren der Astra Control-Bereitstellung.

Bevor Sie beginnen

Amazon Web Services

- Sie sollten die JSON-Datei mit den Anmeldedaten des IAM-Benutzers haben, der das Cluster erstellt hat. "Erfahren Sie, wie ein IAM-Benutzer erstellt wird".
- Astra Control Provisioner ist für Amazon FSX for NetApp ONTAP erforderlich. Wenn Sie Amazon FSX for NetApp ONTAP als Storage-Backend für Ihr EKS-Cluster verwenden möchten, finden Sie in den Informationen zur Astra Control-Bereitstellung im "EKS-Clusteranforderungen".
- (Optional) Wenn Sie angeben müssen `kubectl` Befehlszugriff für ein Cluster auf andere IAM-Benutzer, die nicht der Ersteller des Clusters sind, finden Sie in den Anweisungen unter "Wie erhalte ich Zugriff auf andere IAM-Benutzer und Rollen nach der Cluster-Erstellung in Amazon EKS?".
- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Nutzung mit Amazon Web Services konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Microsoft Azure

- Sie sollten beim Erstellen des Service-Principal die JSON-Datei haben, die die Ausgabe aus der Azure CLI enthält. "Erfahren Sie, wie Sie einen Service-Principal einrichten".

  Außerdem benötigen Sie Ihre Azure Abonnement-ID, wenn Sie sie nicht zur JSON-Datei hinzugefügt haben.

- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Zusammenarbeit mit Microsoft Azure konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Google Cloud

- Sie sollten die Servicekontoschlüsseldatei für ein Servicekonto haben, das über die erforderlichen Berechtigungen verfügt. "Erfahren Sie, wie Sie ein Service-Konto einrichten".
- Wenn das Cluster privat ist, gilt das "Autorisierte Netzwerke" Die Astra Control Service-IP-Adresse muss zugelassen werden:

  52.188.218.166/32

- Wenn Sie NetApp Cloud Volumes ONTAP als Storage-Backend verwenden möchten, müssen Sie Cloud Volumes ONTAP für die Zusammenarbeit mit Google Cloud konfigurieren. Weitere Informationen finden Sie im Cloud Volumes ONTAP "Setup-Dokumentation".

Schritte

1. (Optional) Wenn Sie einen Amazon EKS Cluster hinzufügen oder die Installation und Upgrades von Astra Control Provisioner selbst managen möchten, aktivieren Sie die Astra Control Provisioner-Funktion im Cluster. Siehe "Astra Control Provisioner Aktivieren" Für Enablement-Schritte.
2. Öffnen Sie die Web-UI des Astra Control Service in einem Browser.
3. Wählen Sie im Dashboard **Kubernetes Cluster managen** aus.

   Befolgen Sie die Aufforderungen zum Hinzufügen des Clusters.

4. **Provider**: Wählen Sie Ihren Cloud-Provider aus und geben Sie dann entweder die erforderlichen Anmeldedaten für die Erstellung einer neuen Cloud-Instanz an, oder wählen Sie eine vorhandene Cloud-Instanz aus.

5. **Amazon Web Services**: Geben Sie Details über Ihr Amazon Web Services IAM-Benutzerkonto an, indem Sie eine JSON-Datei hochladen oder den Inhalt dieser JSON-Datei aus Ihrer Zwischenablage einfügen.

   Die JSON-Datei sollte die Anmeldeinformationen des IAM-Benutzers enthalten, der das Cluster erstellt hat.

6. **Microsoft Azure**: Geben Sie Details zu Ihrem Azure Service Principal an, indem Sie eine JSON-Datei hochladen oder den Inhalt dieser JSON-Datei aus Ihrer Zwischenablage einfügen.

   Die JSON-Datei sollte beim Erstellen des Service-Principal die Ausgabe aus der Azure CLI enthalten. Sie können auch Ihre Abonnement-ID angeben, damit sie automatisch in den Astra aufgenommen wird. Andernfalls müssen Sie die ID manuell eingeben, nachdem Sie den JSON bereitgestellt haben.

7. **Google Cloud Platform**: Stellen Sie die Service-Konto-Schlüsseldatei entweder durch das Hochladen der Datei oder durch Einfügen der Inhalte aus Ihrer Zwischenablage bereit.

   Astra Control Service nutzt das Service-Konto, um Cluster zu erkennen, die in der Google Kubernetes Engine ausgeführt werden.

8. **Andere**: Diese Registerkarte ist nur für die Verwendung mit selbst verwalteten Clustern vorgesehen.
   a. **Cloud-Instanzname**: Geben Sie einen Namen für die neue Cloud-Instanz an, die beim Hinzufügen dieses Clusters erstellt wird. Weitere Informationen zu "Cloud-Instanzen".
   b. Wählen Sie **Weiter**.

      Astra Control Service zeigt eine Liste von Clustern an, aus denen Sie auswählen können.

   c. **Cluster**: Wählen Sie einen Cluster aus der Liste aus, der zu Astra Control Service hinzugefügt werden soll.

      > Wenn Sie aus der Liste der Cluster auswählen, achten Sie auf die Spalte **Eligiblity**. Wenn ein Cluster „nicht berechtigt“ oder „teilweise berechtigt“ ist, bewegen Sie den Mauszeiger über den Status, um zu ermitteln, ob ein Problem im Cluster vorliegt. Beispielsweise kann sie erkennen, dass für das Cluster kein Worker Node vorhanden ist.

9. Wählen Sie **Weiter**.
10. (Optional) **Speicher**: Wählen Sie optional die Storage-Klasse aus, die Kubernetes-Anwendungen, die auf diesem Cluster bereitgestellt werden sollen, standardmäßig verwenden sollen.
    a. Um eine neue Standard-Storage-Klasse für den Cluster auszuwählen, aktivieren Sie das Kontrollkästchen **Neue Standard-Storage-Klasse zuweisen**.
    b. Wählen Sie eine neue Standard-Storage-Klasse aus der Liste aus.

Jeder Storage-Service eines Cloud-Providers enthält die folgenden Informationen zu Preis, Performance und Ausfallsicherheit:

- Cloud Volumes Service für Google Cloud: Informationen zu Preis, Performance und Ausfallsicherheit
- Google Persistent Disk: Keine Informationen über Preis, Performance oder Ausfallsicherheit verfügbar
- Azure NetApp Files: Informationen zu Performance und Ausfallsicherheit
- Azure Managed Disks: Es sind weder Preis-, Performance- oder Resilience-Informationen verfügbar
- Amazon Elastic Block Store: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar
- Amazon FSX für NetApp ONTAP: Keine Informationen zu Preis, Performance und Ausfallsicherheit verfügbar
- NetApp Cloud Volumes ONTAP: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar

Jede Storage-Klasse kann einen der folgenden Services nutzen:

- "Cloud Volumes Service für Google Cloud"
- "Google Persistent Disk"
- "Azure NetApp Dateien"
- "Von Azure gemanagte Festplatten"
- "Amazon Elastic Block Store"
- "Amazon FSX für NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

Weitere Informationen zu "Storage-Klassen für Amazon Web Services Cluster". Weitere Informationen zu "Speicherklassen für AKS-Cluster". Weitere Informationen zu "Speicherklassen für GKE-Cluster".

c. Wählen Sie **Weiter**.

d. **Überprüfen und genehmigen**: Überprüfen Sie die Konfigurationsdetails.

e. Wählen Sie **Add**, um den Cluster zu Astra Control Service hinzuzufügen.

Ergebnis

Wenn dies der erste Cluster ist, den Sie für diesen Cloud-Provider hinzugefügt haben, erstellt Astra Control Service einen Objektspeicher für den Cloud-Provider für Backups von Anwendungen, die auf geeigneten Clustern ausgeführt werden. (Wenn Sie nachfolgende Cluster für diesen Cloud-Provider hinzufügen, werden keine weiteren Objektspeicher erstellt.) Wenn Sie eine Standard-Storage-Klasse angegeben haben, setzt Astra Control Service die von Ihnen angegebene Standard-Storage-Klasse ein. Für Cluster, die in Amazon Web Services oder Google Cloud Platform gemanagt werden, erstellt Astra Control Service auch ein Administratorkonto auf dem Cluster. Diese Vorgänge können mehrere Minuten dauern.

Ändern der Standard-Storage-Klasse

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

Ändern Sie die Standard-Storage-Klasse mit Astra Control

Sie können die Standard-Storage-Klasse für ein Cluster aus Astra Control ändern. Wenn Ihr Cluster einen zuvor installierten Speicher-Backend-Service verwendet, können Sie diese Methode möglicherweise nicht verwenden, um die Standard-Speicherklasse zu ändern (die Aktion **default** ist nicht wählbar). In diesem Fall können Sie Ändern Sie die Standard-Storage-Klasse über die Befehlszeile.

Schritte

1. Wählen Sie in der Astra Control Service-UI **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**.

Ändern Sie die Standard-Storage-Klasse über die Befehlszeile

Sie können die Standard-Storage-Klasse für ein Cluster mit Kubernetes-Befehlen ändern. Diese Methode funktioniert unabhängig von der Konfiguration Ihres Clusters.

Schritte

1. Melden Sie sich bei Ihrem Kubernetes Cluster an.
2. Listen Sie die Storage-Klassen in Ihrem Cluster auf:

```
kubectl get storageclass
```

3. Entfernen Sie die Standardbezeichnung aus der Standardspeicherklasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Markieren Sie standardmäßig eine andere Storage-Klasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Bestätigen Sie die neue Standard-Speicherklasse:

```
kubectl get storageclass
```

Hinzufügen eines selbstverwalteten Clusters

Fügen Sie Astra Control Service einen öffentlichen, selbst gemanagten Cluster hinzu

Nach der Einrichtung der Umgebung erstellen Sie sofort einen Kubernetes Cluster und fügen ihn dann dem Astra Control Service hinzu.

Ein selbstverwalteter Cluster ist ein Cluster, den Sie direkt bereitstellen und managen können. Astra Control Service unterstützt selbst gemanagte Cluster, die in einer Public-Cloud-Umgebung ausgeführt werden. Sie können einen selbstverwalteten Cluster zum Astra Control Service hinzufügen, indem Sie ein hochladen `kubeconfig.yaml` Datei: Sie müssen sicherstellen, dass das Cluster die hier aufgeführten Anforderungen erfüllt.

Unterstützte Kubernetes-Distributionen

Mit Astra Control Service können Sie folgende Arten von öffentlichen, selbst gemanagten Clustern managen:

Kubernetes-Distribution	Unterstützte Versionen
Kubernetes (Vorgelagert)	1.27 bis 1.29
Rancher Kubernetes Engine (RKE)	RKE 1: Versionen 1.24.17, 1.25.13, 1.26.8 mit Rancher Manager 2.7.9 RKE 2: Versionen 1.23.16 und 1.24.13 mit Rancher Manager 2.6.13 RKE 2: Versionen 1.24.17, 1.25.14, 1.26.9 mit Rancher Manager 2.7.9
Red hat OpenShift Container Platform	4.12 bis 4.14

Bei diesen Anweisungen wird davon ausgegangen, dass Sie bereits einen selbstverwalteten Cluster erstellt haben.

- Fügen Sie das Cluster zu Astra Control Service hinzu
- Ändern der Standard-Storage-Klasse

Fügen Sie das Cluster zu Astra Control Service hinzu

Nachdem Sie sich beim Astra Control Service angemeldet haben, beginnen Sie zunächst mit dem Verwalten Ihrer Cluster. Bevor Sie Astra Control Service ein Cluster hinzufügen, müssen Sie bestimmte Aufgaben ausführen und sicherstellen, dass das Cluster bestimmte Anforderungen erfüllt.

Bevor Sie beginnen

Ein selbstverwalteter Cluster ist ein Cluster, den Sie direkt bereitstellen und managen können. Astra Control Service unterstützt selbst gemanagte Cluster, die in einer Public-Cloud-Umgebung ausgeführt werden. Die selbstverwalteten Cluster können über Astra Control Provisioner eine Schnittstelle zu NetApp Storage-Services aufbauen. Alternativ können sie über CSI-Treiber (Container Storage Interface) eine Schnittstelle zu Amazon Elastic Block Store (EBS), Azure Managed Disks und Google Persistent Disk erstellen.

Astra Control Service unterstützt selbst gemanagte Cluster, die die folgenden Kubernetes-Distributionen verwenden:

- Red hat OpenShift Container Platform
- Rancher Kubernetes Engine
- Vorgelagerte Kubernetes-Systeme

Ihr Self-Managed-Cluster muss folgende Anforderungen erfüllen:

- Der Cluster muss über das Internet zugänglich sein.
- Wenn Sie Speicher mit CSI-Treibern verwenden oder planen, diese zu verwenden, müssen auf dem Cluster die entsprechenden CSI-Treiber installiert sein. Weitere Informationen zur Verwendung von CSI-Treibern zur Integration von Speicher finden Sie in der Dokumentation Ihres Speicherservices.
- Sie haben Zugriff auf die Cluster-Datei kubeconfig, die nur ein Kontextelement enthält. Folgen "Diese Anweisungen" Um eine kubeconfig-Datei zu erzeugen.
- Wenn Sie den Cluster mit einer kubeconfig-Datei hinzufügen, die auf eine private Zertifizierungsstelle verweist, fügen Sie der folgende Zeile hinzu `cluster` Abschnitt der Datei kubeconfig. So kann Astra Control das Cluster hinzufügen:

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

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

Sie sicher, dass sie die **einzige** Speicherklasse ist, die die Standardanmerkung hat.

- **Konfigurieren Sie einen Volume-Snapshot-Controller und installieren Sie eine Volume-Snapshot-Klasse**: "Installieren Sie einen Volume-Snapshot-Controller" Damit Snapshots in Astra Control erstellt werden können. "Erstellen" Mindestens eine `VolumeSnapshotClass` Einsatz von Astra Trident:

Schritte

1. Wählen Sie im Dashboard **Kubernetes Cluster managen** aus.

   Befolgen Sie die Aufforderungen zum Hinzufügen des Clusters.

2. **Provider**: Wählen Sie den Reiter **andere**, um Details zu Ihrem selbst verwalteten Cluster hinzuzufügen.

   a. **Other**: Geben Sie Details über Ihren selbstverwalteten Cluster durch das Hochladen eines `kubeconfig.yaml` Datei oder durch Einfügen des Inhalts des `kubeconfig.yaml` Datei aus der Zwischenablage.

   > 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. **Credential Name**: Geben Sie einen Namen für die selbstverwalteten Cluster-Zugangsdaten ein, die Sie auf Astra Control hochladen. Standardmäßig wird der Name der Anmeldeinformationen automatisch als Name des Clusters ausgefüllt.
4. **Private Route Identifier**: Dieses Feld ist nur für private Cluster bestimmt.
5. Wählen Sie **Weiter**.
6. (Optional) **Speicher**: Wählen Sie optional die Storage-Klasse aus, die Kubernetes-Anwendungen, die auf diesem Cluster bereitgestellt werden sollen, standardmäßig verwenden sollen.

   a. Um eine neue Standard-Storage-Klasse für den Cluster auszuwählen, aktivieren Sie das Kontrollkästchen **Neue Standard-Storage-Klasse zuweisen**.

   b. Wählen Sie eine neue Standard-Storage-Klasse aus der Liste aus.

Jeder Storage-Service eines Cloud-Providers enthält die folgenden Informationen zu Preis, Performance und Ausfallsicherheit:

- Cloud Volumes Service für Google Cloud: Informationen zu Preis, Performance und Ausfallsicherheit
- Google Persistent Disk: Keine Informationen über Preis, Performance oder Ausfallsicherheit verfügbar
- Azure NetApp Files: Informationen zu Performance und Ausfallsicherheit
- Azure Managed Disks: Es sind weder Preis-, Performance- oder Resilience-Informationen verfügbar
- Amazon Elastic Block Store: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar
- Amazon FSX für NetApp ONTAP: Keine Informationen zu Preis, Performance und Ausfallsicherheit verfügbar
- NetApp Cloud Volumes ONTAP: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar

Jede Storage-Klasse kann einen der folgenden Services nutzen:

- "Cloud Volumes Service für Google Cloud"
- "Google Persistent Disk"
  - "Azure NetApp Dateien"
  - "Von Azure gemanagte Festplatten"
  - "Amazon Elastic Block Store"
  - "Amazon FSX für NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

  Weitere Informationen zu "Storage-Klassen für Amazon Web Services Cluster". Weitere Informationen zu "Speicherklassen für AKS-Cluster". Weitere Informationen zu "Speicherklassen für GKE-Cluster".

c. Wählen Sie **Weiter**.

d. **Überprüfen und genehmigen**: Überprüfen Sie die Konfigurationsdetails.

e. Wählen Sie **Add**, um den Cluster zu Astra Control Service hinzuzufügen.

Ändern der Standard-Storage-Klasse

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

Ändern Sie die Standard-Storage-Klasse mit Astra Control

Sie können die Standard-Storage-Klasse für ein Cluster aus Astra Control ändern. Wenn Ihr Cluster einen zuvor installierten Speicher-Backend-Service verwendet, können Sie diese Methode möglicherweise nicht verwenden, um die Standard-Speicherklasse zu ändern (die Aktion **default** ist nicht wählbar). In diesem Fall können Sie Ändern Sie die Standard-Storage-Klasse über die Befehlszeile.

Schritte

1. Wählen Sie in der Astra Control Service-UI **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**.

Ändern Sie die Standard-Storage-Klasse über die Befehlszeile

Sie können die Standard-Storage-Klasse für ein Cluster mit Kubernetes-Befehlen ändern. Diese Methode funktioniert unabhängig von der Konfiguration Ihres Clusters.

Schritte

1. Melden Sie sich bei Ihrem Kubernetes Cluster an.
2. Listen Sie die Storage-Klassen in Ihrem Cluster auf:

```
kubectl get storageclass
```

3. Entfernen Sie die Standardbezeichnung aus der Standardspeicherklasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Markieren Sie standardmäßig eine andere Storage-Klasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Bestätigen Sie die neue Standard-Speicherklasse:

```
kubectl get storageclass
```

Fügen Sie Astra Control Service einen privaten, selbst gemanagten Cluster hinzu

Nach der Einrichtung der Umgebung erstellen Sie sofort einen Kubernetes Cluster und fügen ihn dann dem Astra Control Service hinzu.

Ein selbstverwalteter Cluster ist ein Cluster, den Sie direkt bereitstellen und managen können. Astra Control Service unterstützt selbst gemanagte Cluster, die in einer Public-Cloud-Umgebung ausgeführt werden. Sie können einen selbstverwalteten Cluster zum Astra Control Service hinzufügen, indem Sie ein hochladen

`kubeconfig.yaml` Datei: Sie müssen sicherstellen, dass das Cluster die hier aufgeführten Anforderungen erfüllt.

Unterstützte Kubernetes-Distributionen

Mit Astra Control Service können Sie folgende Arten von privaten, selbst gemanagten Clustern managen:

Kubernetes-Distribution	Unterstützte Versionen
Kubernetes (Vorgelagert)	1.27 bis 1.29
Rancher Kubernetes Engine (RKE)	RKE 1: Versionen 1.24.17, 1.25.13, 1.26.8 mit Rancher Manager 2.7.9 RKE 2: Versionen 1.23.16 und 1.24.13 mit Rancher Manager 2.6.13 RKE 2: Versionen 1.24.17, 1.25.14, 1.26.9 mit Rancher Manager 2.7.9
Red hat OpenShift Container Platform	4.12 bis 4.14

In diesen Anweisungen wird davon ausgegangen, dass Sie bereits einen privaten Cluster erstellt und eine sichere Methode für den Remote-Zugriff darauf vorbereitet haben.

Führen Sie die folgenden Aufgaben aus, um Ihren privaten Cluster zum Astra Control Service hinzuzufügen:

1. Astra Connector Installieren
2. Einrichtung von persistentem Storage
3. selbst gemanagten Cluster zum Astra Control Service hinzu

Astra Connector Installieren

Bevor Sie einen privaten Cluster hinzufügen, müssen Sie Astra Connector im Cluster installieren, damit Astra Control damit kommunizieren kann. Siehe "Installieren Sie die vorherige Version von Astra Connector für private Cluster, die mit nicht-Kubernetes-nativen Workflows gemanagt werden" Weitere Anweisungen.

Einrichtung von persistentem Storage

Konfigurieren Sie persistenten Storage für das Cluster. In der Dokumentation „erste Schritte“ finden Sie weitere Informationen zum Konfigurieren von persistentem Storage:

- "Microsoft Azure mit Azure NetApp Files einrichten"
- "Richten Sie Microsoft Azure mit von Azure gemanagten Festplatten ein"
- "Einrichten von Amazon Web Services"
- "Google Cloud einrichten"

Fügen Sie den privaten, selbst gemanagten Cluster zum Astra Control Service hinzu

Sie können den privaten Cluster jetzt dem Astra Control Service hinzufügen.

Bevor Sie beginnen

Ein selbstverwalteter Cluster ist ein Cluster, den Sie direkt bereitstellen und managen können. Astra Control Service unterstützt selbst gemanagte Cluster, die in einer Public-Cloud-Umgebung ausgeführt werden. Die selbstverwalteten Cluster können über Astra Control Provisioner eine Schnittstelle zu NetApp Storage-Services aufbauen. Alternativ können sie über CSI-Treiber (Container Storage Interface) eine Schnittstelle zu Amazon Elastic Block Store (EBS), Azure Managed Disks und Google Persistent Disk erstellen.

Astra Control Service unterstützt selbst gemanagte Cluster, die die folgenden Kubernetes-Distributionen verwenden:

- Red hat OpenShift Container Platform
- Rancher Kubernetes Engine
- Vorgelagerte Kubernetes-Systeme

Ihr Self-Managed-Cluster muss folgende Anforderungen erfüllen:

- Der Cluster muss über das Internet zugänglich sein.
- Wenn Sie Speicher mit CSI-Treibern verwenden oder planen, diese zu verwenden, müssen auf dem Cluster die entsprechenden CSI-Treiber installiert sein. Weitere Informationen zur Verwendung von CSI-Treibern zur Integration von Speicher finden Sie in der Dokumentation Ihres Speicherservices.
- Sie haben Zugriff auf die Cluster-Datei kubeconfig, die nur ein Kontextelement enthält. Folgen "Diese Anweisungen" Um eine kubeconfig-Datei zu erzeugen.
- Wenn Sie den Cluster mit einer kubeconfig-Datei hinzufügen, die auf eine private Zertifizierungsstelle verweist, fügen Sie der folgende Zeile hinzu `cluster` Abschnitt der Datei kubeconfig. So kann Astra Control das Cluster hinzufügen:

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

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

Sie sicher, dass sie die **einzige** Speicherklasse ist, die die Standardanmerkung hat.

- **Konfigurieren Sie einen Volume-Snapshot-Controller und installieren Sie eine Volume-Snapshot-Klasse**: "Installieren Sie einen Volume-Snapshot-Controller" Damit Snapshots in Astra Control erstellt werden können. "Erstellen" Mindestens eine `VolumeSnapshotClass` Einsatz von Astra Trident:

Schritte

1. Wählen Sie im Dashboard **Kubernetes Cluster managen** aus.

   Befolgen Sie die Aufforderungen zum Hinzufügen des Clusters.

2. **Provider**: Wählen Sie den Reiter **andere**, um Details zu Ihrem selbst verwalteten Cluster hinzuzufügen.
3. **Other**: Geben Sie Details über Ihren selbstverwalteten Cluster durch das Hochladen eines `kubeconfig.yaml` Datei oder durch Einfügen des Inhalts des `kubeconfig.yaml` Datei aus der Zwischenablage.

   > Wenn Sie Ihre eigenen erstellen `kubeconfig` Datei, Sie sollten nur ein **ein**-Kontext -Element darin definieren. Siehe "Diese Anweisungen" Weitere Informationen zum Erstellen `kubeconfig` Dateien:

4. **Credential Name**: Geben Sie einen Namen für die selbstverwalteten Cluster-Zugangsdaten ein, die Sie auf Astra Control hochladen. Standardmäßig wird der Name der Anmeldeinformationen automatisch als Name des Clusters ausgefüllt.
5. **Private Route Identifier**: Geben Sie die private Route Identifier ein, die Sie vom Astra Connector erhalten können. Wenn Sie den Astra Connector über die abfragen `kubectl get astraconnector -n astra-connector` Die Kennung der privaten Route wird als bezeichnet `ASTRACONNECTORID`.

   > Die Private-Route-ID ist der Name, der dem Astra Connector zugeordnet ist. Damit kann ein privates Kubernetes-Cluster von Astra gemanagt werden. In diesem Kontext ist ein privates Cluster ein Kubernetes-Cluster, das seinen API-Server nicht zum Internet bereitstellt.

6. Wählen Sie **Weiter**.
7. (Optional) **Speicher**: Wählen Sie optional die Storage-Klasse aus, die Kubernetes-Anwendungen, die auf diesem Cluster bereitgestellt werden sollen, standardmäßig verwenden sollen.
   a. Um eine neue Standard-Storage-Klasse für den Cluster auszuwählen, aktivieren Sie das Kontrollkästchen **Neue Standard-Storage-Klasse zuweisen**.
   b. Wählen Sie eine neue Standard-Storage-Klasse aus der Liste aus.

Jeder Storage-Service eines Cloud-Providers enthält die folgenden Informationen zu Preis, Performance und Ausfallsicherheit:

- Cloud Volumes Service für Google Cloud: Informationen zu Preis, Performance und Ausfallsicherheit
- Google Persistent Disk: Keine Informationen über Preis, Performance oder Ausfallsicherheit verfügbar
- Azure NetApp Files: Informationen zu Performance und Ausfallsicherheit
- Azure Managed Disks: Es sind weder Preis-, Performance- oder Resilience-Informationen verfügbar
- Amazon Elastic Block Store: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar
- Amazon FSX für NetApp ONTAP: Keine Informationen zu Preis, Performance und Ausfallsicherheit verfügbar
- NetApp Cloud Volumes ONTAP: Keine Informationen zu Preis, Performance oder Ausfallsicherheit verfügbar

Jede Storage-Klasse kann einen der folgenden Services nutzen:

- "Cloud Volumes Service für Google Cloud"
- "Google Persistent Disk"
- "Azure NetApp Dateien"
- "Von Azure gemanagte Festplatten"
- "Amazon Elastic Block Store"
- "Amazon FSX für NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

Weitere Informationen zu "Storage-Klassen für Amazon Web Services Cluster". Weitere Informationen zu "Speicherklassen für AKS-Cluster". Weitere Informationen zu "Speicherklassen für GKE-Cluster".

c. Wählen Sie **Weiter**.

d. **Überprüfen und genehmigen**: Überprüfen Sie die Konfigurationsdetails.

e. Wählen Sie **Add**, um den Cluster zu Astra Control Service hinzuzufügen.

Ändern der Standard-Storage-Klasse

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

Ändern Sie die Standard-Storage-Klasse mit Astra Control

Sie können die Standard-Storage-Klasse für ein Cluster aus Astra Control ändern. Wenn Ihr Cluster einen zuvor installierten Speicher-Backend-Service verwendet, können Sie diese Methode möglicherweise nicht verwenden, um die Standard-Speicherklasse zu ändern (die Aktion **default** ist nicht wählbar). In diesem Fall können Sie Ändern Sie die Standard-Storage-Klasse über die Befehlszeile.

Schritte

1. Wählen Sie in der Astra Control Service-UI **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**.

Ändern Sie die Standard-Storage-Klasse über die Befehlszeile

Sie können die Standard-Storage-Klasse für ein Cluster mit Kubernetes-Befehlen ändern. Diese Methode funktioniert unabhängig von der Konfiguration Ihres Clusters.

Schritte

1. Melden Sie sich bei Ihrem Kubernetes Cluster an.
2. Listen Sie die Storage-Klassen in Ihrem Cluster auf:

```
kubectl get storageclass
```

3. Entfernen Sie die Standardbezeichnung aus der Standardspeicherklasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Markieren Sie standardmäßig eine andere Storage-Klasse. Ersetzen Sie <SC_NAME> durch den Namen der Speicherklasse:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Bestätigen Sie die neue Standard-Speicherklasse:

```
kubectl get storageclass
```

Prüfen Sie die Astra Trident Version

Wenn Sie einen selbst gemanagten Cluster hinzufügen möchten, der Astra Control Provisioner oder Astra Trident für Storage-Services verwendet, müssen Sie sicherstellen, dass die installierte Version von Astra Trident 23.10 oder aktuell ist.

Schritte

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

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

Wenn Astra Trident installiert ist, wird die Ausgabe wie folgt ausgegeben:

```
NAME      VERSION
trident   24.02.0
```

Wenn Astra Trident nicht installiert ist, wird die Ausgabe wie folgt angezeigt:

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

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

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

4. Prüfen Sie, ob die Storage-Klassen die unterstützten Astra Trident Treiber verwenden. Der bereitstellungsname sollte lauten `csi.trident.netapp.io`. Im folgenden Beispiel finden Sie weitere Informationen:

```
kubectl get sc
```

Beispielantwort:

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

Erstellen Sie eine kubeconfig-Datei

Sie können dem Astra Control Service ein Cluster mithilfe einer kubeconfig-Datei hinzufügen. Je nach dem Typ des Clusters, den Sie hinzufügen möchten, müssen Sie möglicherweise manuell eine kubeconfig-Datei für Ihr Cluster mithilfe bestimmter Schritte erstellen.

- Erstellen Sie eine kubeconfig-Datei für Amazon EKS-Cluster
- Erstellen Sie eine kubeconfig-Datei für Red hat OpenShift Service on AWS (ROSA) Cluster
- Erstellen Sie eine kubeconfig-Datei für andere Cluster-Typen

Erstellen Sie eine kubeconfig-Datei für Amazon EKS-Cluster

Befolgen Sie diese Anweisungen, um eine kubeconfig-Datei und ein permanentes Token-Geheimnis für Amazon EKS-Cluster zu erstellen. Für Cluster, die in EKS gehostet werden, ist ein permanenter Tokenschlüssel erforderlich.

Schritte

1. Befolgen Sie die Anweisungen in der Amazon-Dokumentation, um eine kubeconfig-Datei zu erstellen:

   "Erstellen oder Aktualisieren einer kubeconfig-Datei für einen Amazon EKS-Cluster"

2. Erstellen Sie ein Service-Konto wie folgt:

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

      Passen Sie den Namen des Servicekontos nach Bedarf an. Der Namespace `kube-system` Ist für diese Schritte erforderlich. Wenn Sie hier den Namen des Servicekontos ändern, sollten Sie die gleichen Änderungen in den folgenden Schritten anwenden.

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

+

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. Wenden Sie das Servicekonto an:

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

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

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

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

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

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

6. Erstellen Sie eine Geheimdatei für das Dienstkonto-Token mit dem Namen `astracontrol-secret.yaml`.

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

```yaml
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. Wenden Sie den Token-Schlüssel an:

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

8. Rufen Sie den Token-Schlüssel ab:

```
kubectl get secret astra-admin-account -n kube-system -o
jsonpath='{.data.token}' | base64 -d
```

9. Ersetzen Sie den `user` Abschnitt der AWS EKS kubeconfig-Datei mit dem Token, wie im folgenden Beispiel gezeigt:

```
user:
    token: k8s-aws-
v1.aHR0cHM6Ly9zdHMudXMtd2VzdC0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ
XJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQ
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBM1JEWDdKU0haWU9LSEQ2SyUyRjIwMjMwN
DAzJTJGdXMtd2VzdC0yJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzA0M
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQ
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YjU4ZWM0NzdiM2NkZGYxNGRhNzU4MGI2Z
WQ2zY2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

Erstellen Sie eine kubeconfig-Datei für Red hat OpenShift Service on AWS (ROSA) Cluster

Befolgen Sie diese Anweisungen, um eine kubeconfig-Datei für Red hat OpenShift Service on AWS (ROSA)-Cluster zu erstellen.

Schritte

1. Melden Sie sich beim ROSA-Cluster an.
2. Service-Konto erstellen:

```
oc create sa astracontrol-service-account
```

3. Cluster-Rolle hinzufügen:

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-
service-account
```

4. Erstellen Sie mithilfe des folgenden Beispiels eine geheime Konfigurationsdatei für das Dienstkonto:

```
<strong>secret-astra-sa.yaml</strong>
```

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

5. Erstellen Sie das Geheimnis:

```
oc create -f secret-astra-sa.yaml
```

6. Bearbeiten Sie das von Ihnen erstellte Dienstkonto, und fügen Sie dem den geheimen Namen des Astra Control-Dienstkontos hinzu `secrets` Abschnitt:

```
oc edit sa astracontrol-service-account
```

```yaml
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-dvfcd
kind: ServiceAccount
metadata:
  creationTimestamp: "2023-08-04T04:18:30Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "169770"
  uid: 965fa151-923f-4fbd-9289-30cad15998ac
secrets:
- name: astracontrol-service-account-dockercfg-dvfcd
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```

7. 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-dvfcd"},
{ "name": "secret-astracontrol-service-account"}
]
```

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

8. 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
```

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Erstellen Sie eine kubeconfig-Datei für andere Cluster-Typen

Befolgen Sie diese Anweisungen, um eine begrenzte oder erweiterte Kubeconfig-Datei für Rancher-, Upstream-Kubernetes- und Red hat OpenShift-Cluster zu erstellen.

Für Cluster, die mit kubeconfig gemanagt werden, können Sie optional eine Administratorrolle mit eingeschränkter Berechtigung oder erweiterten Berechtigungen für Astra Control Service erstellen.

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

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

Bevor Sie beginnen

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

- A "Unterstützte Version" Von kubectl ist installiert.
- Kubectl Zugriff auf den Cluster, den Sie mit Astra Control Service hinzufügen und managen möchten

> Für dieses Verfahren benötigen Sie keinen kubectl-Zugriff auf den Cluster, auf dem Astra Control Service ausgeführt wird.

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

Schritte

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

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

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

   b. Wenden Sie das Servicekonto an:

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

2. Erstellen Sie eine der folgenden Clusterrollen mit ausreichenden Berechtigungen für ein Cluster, das von

Astra Control gemanagt werden kann:

Eingeschränkte Cluster-Rolle

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

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

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

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

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

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

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

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

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

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

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

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

c. Wenden Sie die Cluster-Rolle an:

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

Erweiterte Cluster-Rolle

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

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

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

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

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

b. Wenden Sie die Cluster-Rolle an:

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

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

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

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

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

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

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

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

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

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

   b. Wenden Sie den Token-Schlüssel an:

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

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

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

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

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

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

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

Das Ende der Ausgabe sollte wie folgt aussehen:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

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

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

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

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  *-o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
```

```
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

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

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

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```