Astra Connector installieren, um Cluster zu managen

Astra Control Service

NetApp
April 24, 2024

Inhalt

Astra Connector installieren, um Cluster zu managen 1
Installieren Sie die vorherige Version von Astra Connector 1
(Tech Preview) Installieren Sie den deklarativen Kubernetes Astra Connector 4

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.

```
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
```

```
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.