Anforderungen

Kubernetes clusters

NetApp
December 15, 2022

Inhaltsverzeichnis

Anforderungen

Anforderungen an Kubernetes-Cluster in AWS

Sie können verwaltete Amazon Elastic Kubernetes Service (EKS) Cluster oder automatisierte Kubernetes-Cluster auf AWS zu BlueXP hinzufügen. Bevor Sie die Cluster zu BlueXP hinzufügen können, müssen Sie sicherstellen, dass die folgenden Anforderungen erfüllt sind.

In diesem Thema wird *Kubernetes Cluster* verwendet, wobei die Konfiguration für EKS und selbst gemanagte Kubernetes Cluster identisch ist. Der Cluster-Typ wird bei unterschiedlich der Konfiguration angegeben.

Anforderungen

Astra Trident

Eine der vier aktuellsten Versionen von Astra Trident ist erforderlich. Sie können Astra Trident direkt von BlueXP installieren oder aktualisieren. Sollten Sie "Prüfen Sie die Voraussetzungen" Vor der Installation von Astra Trident:

Cloud Volumes ONTAP

Cloud Volumes ONTAP für AWS muss als Back-End Storage für den Cluster eingerichtet werden. "In der Astra Trident Dokumentation finden Sie die Konfigurationsschritte".

BlueXP Connector

Ein Connector muss in AWS mit den erforderlichen Berechtigungen ausgeführt werden. Weitere Informationen finden Sie unten.

Netzwerk-Konnektivität

Zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und Cloud Volumes ONTAP ist eine Netzwerkverbindung erforderlich. Weitere Informationen finden Sie unten.

RBAC-Autorisierung

Die BlueXP Connector-Rolle muss für jeden Kubernetes-Cluster autorisiert sein. Weitere Informationen finden Sie unten.

Bereiten Sie einen Konnektor vor

Für die Erkennung und das Management von Kubernetes-Clustern ist in AWS ein BlueXP Connector erforderlich. Sie müssen einen neuen Konnektor erstellen oder einen vorhandenen Konnektor verwenden, der über die erforderlichen Berechtigungen verfügt.

Erstellen Sie einen neuen Konnektor

Folgen Sie den Schritten in einem der nachfolgenden Links.

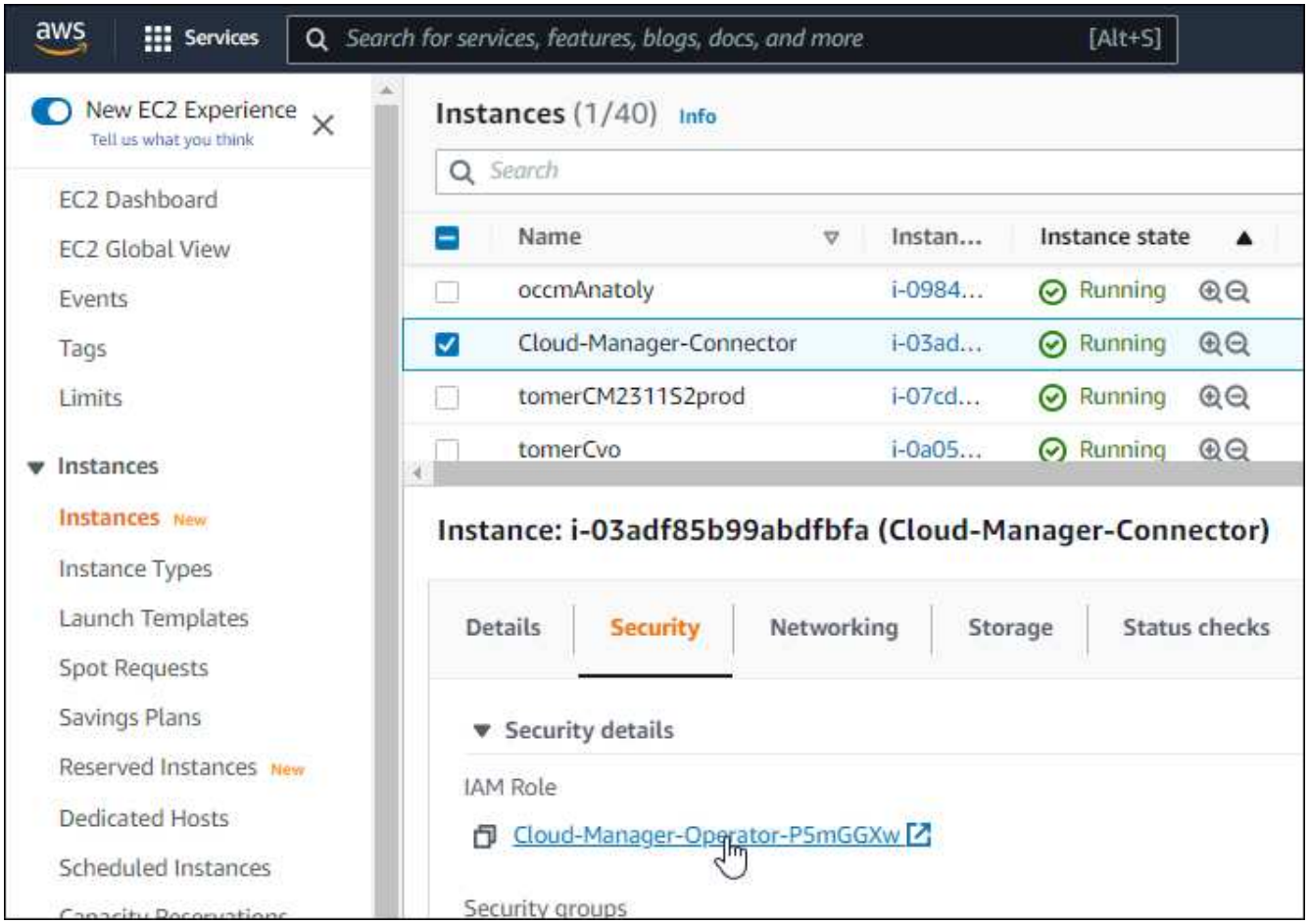
- "Erstellen Sie einen Connector von BlueXP" (Empfohlen)
- "Erstellen Sie einen Connector aus dem AWS Marketplace"
- "Installieren Sie den Connector auf einem vorhandenen Linux-Host in AWS"

Fügen Sie die erforderlichen Berechtigungen einem vorhandenen Konnektor hinzu

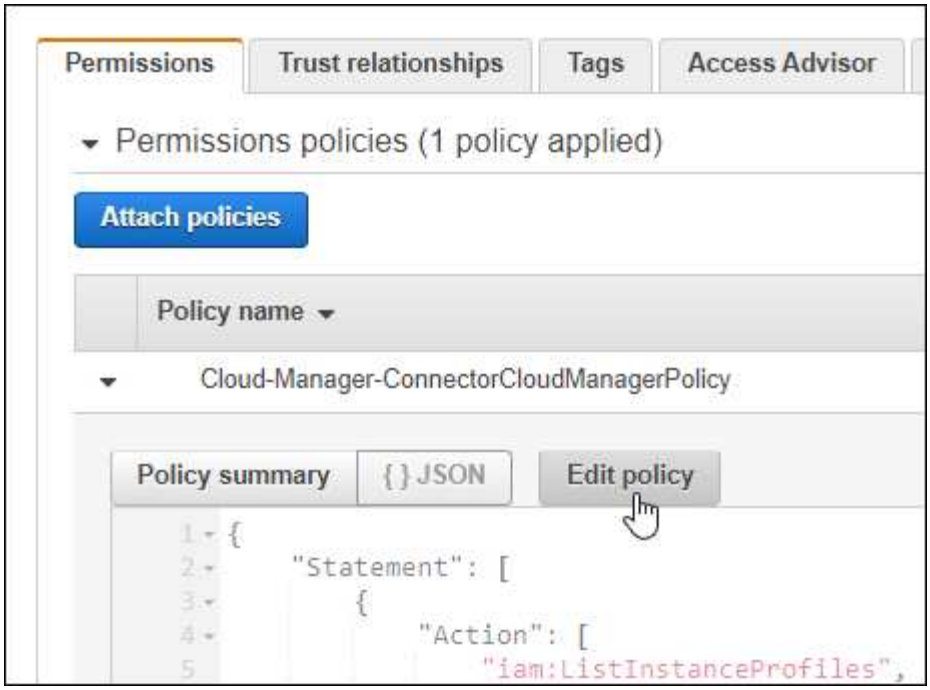
Ab Version 3.9.13 enthalten alle neu erstellten _Connectors drei neue AWS Berechtigungen, die das Erkennen und Managen von Kubernetes-Clustern ermöglichen. Wenn Sie vor dieser Version einen Connector erstellt haben, müssen Sie die vorhandene Richtlinie für die IAM-Rolle des Connectors ändern, um die Berechtigungen bereitzustellen.

Schritte

1. Gehen Sie zur AWS Konsole und öffnen Sie den EC2 Service.
2. Wählen Sie die Connector-Instanz aus, klicken Sie auf **Sicherheit** und klicken Sie auf den Namen der IAM-Rolle, um die Rolle im IAM-Service anzuzeigen.



3. Erweitern Sie auf der Registerkarte **Berechtigungen** die Richtlinie und klicken Sie auf **Richtlinie bearbeiten**.



4. Klicken Sie auf **JSON** und fügen Sie unter dem ersten Satz von Aktionen die folgenden Berechtigungen hinzu:

   - ec2:DescribeRegionen
   - eks:ListClusters
   - eks:DescribeCluster
   - iam:GetInstanceProfile

   "Zeigen Sie das vollständige JSON-Format für die Richtlinie an"

5. Klicken Sie auf **Richtlinie überprüfen** und dann auf **Änderungen speichern**.

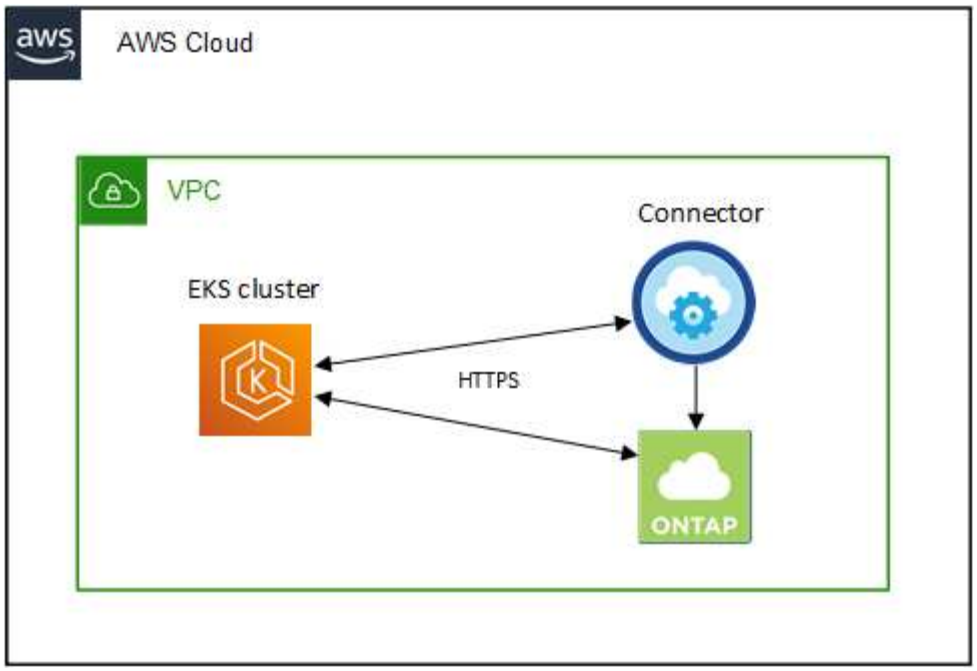
Netzwerkanforderungen prüfen

Sie müssen für die Netzwerkverbindung zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und dem Cloud Volumes ONTAP-System sorgen, das dem Cluster Back-End-Storage bereitstellt.

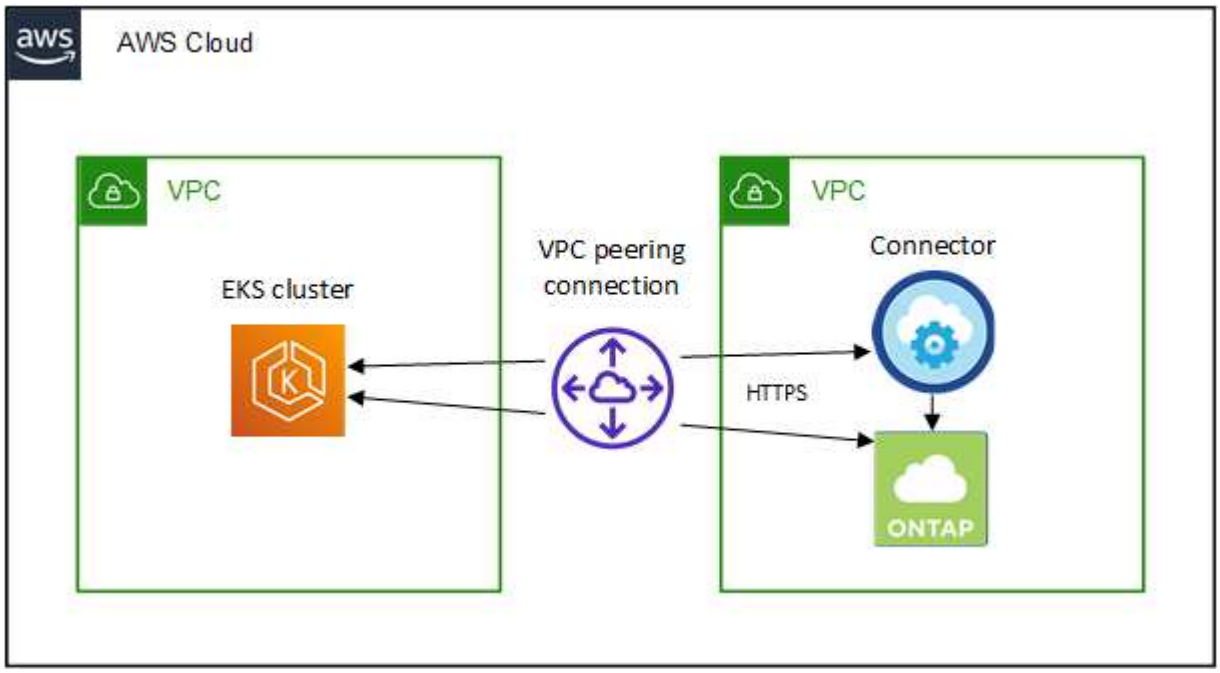
- Jeder Kubernetes-Cluster muss über eine eingehende Verbindung vom Connector verfügen
- Der Connector muss über Port 443 eine ausgehende Verbindung zu jedem Kubernetes-Cluster haben

Die einfachste Möglichkeit für diese Konnektivität ist die Implementierung von Connector und Cloud Volumes ONTAP in derselben VPC wie der Kubernetes-Cluster. Andernfalls müssen Sie eine VPC-Peering-Verbindung zwischen den verschiedenen VPCs einrichten.

In diesem Beispiel wird jede Komponente in derselben VPC angezeigt.



Ein weiteres Beispiel zeigt einen EKS-Cluster, der in einem anderen VPC ausgeführt wird. In diesem Beispiel stellt VPC Peering eine Verbindung zwischen der VPC für das EKS-Cluster und der VPC für den Connector und Cloud Volumes ONTAP her.



Einrichtung der RBAC-Autorisierung

Sie müssen die Connector-Rolle auf jedem Kubernetes-Cluster autorisieren, damit der Connector einen Cluster ermitteln und verwalten kann.

Es ist eine andere Autorisierung erforderlich, um andere Funktionen zu aktivieren.

Backup und Restore

Für Backup und Restore ist nur eine Grundautorisierung erforderlich.

Fügen Sie Speicherklassen hinzu

Erweiterte Autorisierung ist erforderlich, um Speicherklassen mithilfe von BlueXP hinzuzufügen und den Cluster auf Änderungen am Backend zu überwachen.

Installieren Sie Astra Trident

Zur Installation von Astra Trident müssen Sie für BlueXP die vollständige Autorisierung bereitstellen.

Bei der Installation von Astra Trident installiert BlueXP das Astra Trident Back-End und das Kubernetes Secret, das die Zugangsdaten enthält, die Astra Trident zur Kommunikation mit dem Storage-Cluster benötigt.

Schritte

1. Erstellen Sie eine Cluster-Rolle und Rollenbindung.
   a. Erstellen Sie eine YAML-Datei, die den folgenden Text enthält, der auf Ihren Autorisierungsanforderungen basiert.

Backup/Restore

Fügen Sie eine grundlegende Autorisierung hinzu, um Backup und Restore für Kubernetes-Cluster zu ermöglichen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
        - ''
      resources:
        - namespaces
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumes
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - pods
        - pods/exec
      verbs:
        - get
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumeclaims
      verbs:
        - list
        - create
        - watch
    - apiGroups:
        - storage.k8s.io
      resources:
        - storageclasses
      verbs:
```

```
            - list
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentbackends
      verbs:
            - list
            - watch
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentorchestrators
      verbs:
            - get
            - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: Group
      name: cloudmanager-access-group
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Speicherklassen

Fügen Sie erweiterte Berechtigungen hinzu, um Speicherklassen mithilfe von BlueXP hinzuzufügen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
            - ''
      resources:
            - secrets
            - namespaces
            - persistentvolumeclaims
            - persistentvolumes
```

```
          - pods
          - pods/exec
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
    - apiGroups:
          - storage.k8s.io
      resources:
          - storageclasses
      verbs:
          - get
          - create
          - list
          - watch
          - delete
          - patch
    - apiGroups:
          - trident.netapp.io
      resources:
          - tridentbackends
          - tridentorchestrators
          - tridentbackendconfigs
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: Group
      name: cloudmanager-access-group
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Installation Von Trident

Über die Befehlszeile erhalten Sie die vollständige Autorisierung, und BlueXP kann Astra Trident installieren.

```
eksctl create iamidentitymapping --cluster < > --region < > --arn
< > --group "system:masters" --username
system:node:{{EC2PrivateDNSName}}
```

b. Wenden Sie die Konfiguration auf ein Cluster an.

```
kubectl apply -f <file-name>
```

2. Erstellen Sie eine Identitätszuordnung zur Berechtigungsgruppe.

Verwenden Sie eksctl

Verwenden Sie eksctl, um eine IAM-Identitätszuordnung zwischen einem Cluster und der IAM-Rolle für den BlueXP Connector zu erstellen.

"Die vollständige Anleitung finden Sie in der eksctl-Dokumentation".

Im Folgenden finden Sie ein Beispiel.

```
eksctl create iamidentitymapping --cluster <eksCluster> --region
<us-east-2> --arn <ARN of the Connector IAM role> --group
cloudmanager-access-group --username
system:node:{{EC2PrivateDNSName}}
```

Bearbeiten von aws-auth

Bearbeiten Sie die aws-auth ConfigMap direkt, um dem BlueXP Connector den RBAC-Zugriff auf die IAM-Rolle hinzuzufügen.

"Vollständige Anweisungen finden Sie in der AWS EKS-Dokumentation".

Im Folgenden finden Sie ein Beispiel.

```
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - cloudmanager-access-group
      rolearn: <ARN of the Connector IAM role>
     username: system:node:{{EC2PrivateDNSName}}
kind: ConfigMap
metadata:
  creationTimestamp: "2021-09-30T21:09:18Z"
  name: aws-auth
  namespace: kube-system
  resourceVersion: "1021"
  selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
  uid: dcc31de5-3838-11e8-af26-02e00430057c
```

Anforderungen an Kubernetes Cluster in Azure

Verwaltete Azure Kubernetes-Cluster (AKS) und automatisierte Kubernetes-Cluster in Azure können mithilfe von BlueXP hinzugefügt und gemanagt werden. Bevor Sie die Cluster zu BlueXP hinzufügen können, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

In diesem Thema wird *Kubernetes Cluster* verwendet, wobei die Konfiguration für AKS und selbst gemanagte Kubernetes Cluster identisch ist. Der Cluster-Typ wird bei unterschiedlich der Konfiguration angegeben.

Anforderungen

Astra Trident

Eine der vier aktuellsten Versionen von Astra Trident ist erforderlich. Sie können Astra Trident direkt von BlueXP installieren oder aktualisieren. Sollten Sie "Prüfen Sie die Voraussetzungen" Vor der Installation von Astra Trident:

Cloud Volumes ONTAP

Cloud Volumes ONTAP muss als Back-End Storage für den Cluster eingerichtet werden. "In der Astra Trident Dokumentation finden Sie die Konfigurationsschritte".

BlueXP Connector

In Azure muss ein Connector mit den erforderlichen Berechtigungen ausgeführt werden. Weitere Informationen finden Sie unten.

Netzwerk-Konnektivität

Zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und Cloud Volumes ONTAP ist eine Netzwerkverbindung erforderlich. Weitere Informationen finden Sie unten.

RBAC-Autorisierung

BlueXP unterstützt RBAC-fähige Cluster mit und ohne Active Directory. Die BlueXP Connector-Rolle muss für jeden Azure-Cluster autorisiert sein. Weitere Informationen finden Sie unten.

Bereiten Sie einen Konnektor vor

Für das Erkennen und Managen von Kubernetes-Clustern ist ein BlueXP Connector in Azure erforderlich. Sie müssen einen neuen Konnektor erstellen oder einen vorhandenen Konnektor verwenden, der über die erforderlichen Berechtigungen verfügt.

Erstellen Sie einen neuen Konnektor

Folgen Sie den Schritten in einem der nachfolgenden Links.

- "Erstellen Sie einen Connector von BlueXP" (Empfohlen)
- "Erstellen Sie einen Connector aus dem Azure Marketplace"
- "Installieren Sie den Connector auf einem vorhandenen Linux-Host"

Fügen Sie die erforderlichen Berechtigungen einem bestehenden Connector hinzu (um ein verwaltetes AKS-Cluster zu ermitteln)

Wenn Sie einen verwalteten AKS-Cluster ermitteln möchten, müssen Sie möglicherweise die benutzerdefinierte Rolle ändern, damit der Connector die Berechtigungen bereitstellen kann.

Schritte

1. Identifizieren Sie die Rolle, die der virtuellen Konnektor-Maschine zugewiesen ist:
   a. Öffnen Sie im Azure-Portal den Virtual Machines-Service.

b. Wählen Sie die virtuelle Verbindungsmaschine aus.
c. Wählen Sie unter Einstellungen **Identität** aus.
d. Klicken Sie auf **Azure Rollenzuweisungen**.
e. Notieren Sie sich die benutzerdefinierte Rolle, die der virtuellen Connector-Maschine zugewiesen ist.

2. Aktualisieren der benutzerdefinierten Rolle:
   a. Öffnen Sie im Azure-Portal Ihr Azure-Abonnement.
   b. Klicken Sie auf **Zugriffskontrolle (IAM) > Rollen**.
   c. Klicken Sie auf die Ellipsen (…) für die benutzerdefinierte Rolle und dann auf **Bearbeiten**.
   d. Klicken Sie auf JSON und fügen Sie die folgenden Berechtigungen hinzu:

```
"Microsoft.ContainerService/managedClusters/listClusterUserCredential
/action"
"Microsoft.ContainerService/managedClusters/read"
```

   e. Klicken Sie auf **Review + Update** und dann auf **Update**.

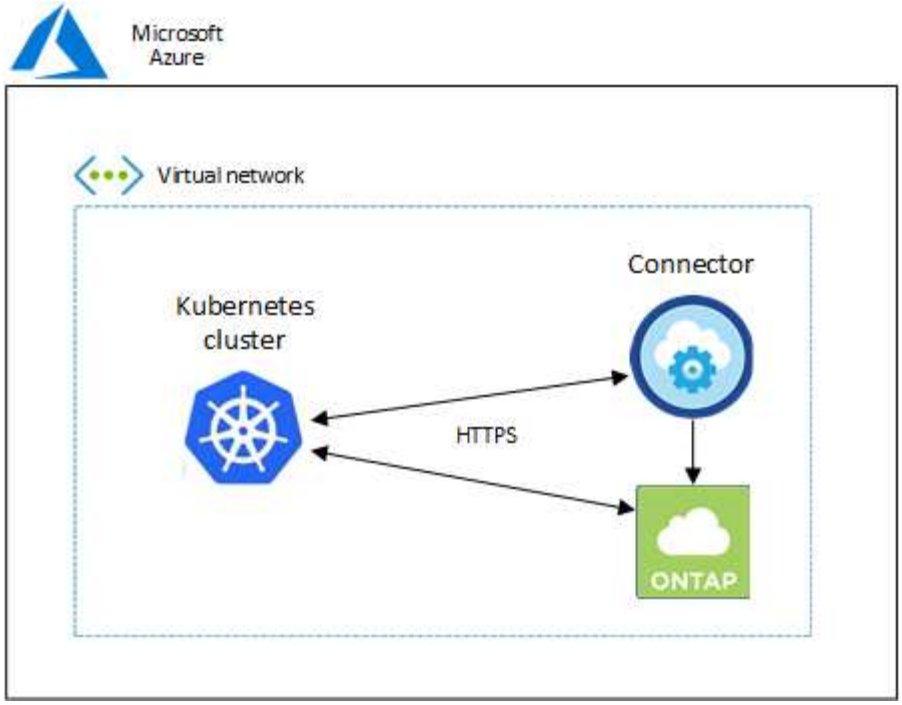
Netzwerkanforderungen prüfen

Sie müssen für die Netzwerkverbindung zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und dem Cloud Volumes ONTAP-System sorgen, das dem Cluster Back-End-Storage bereitstellt.

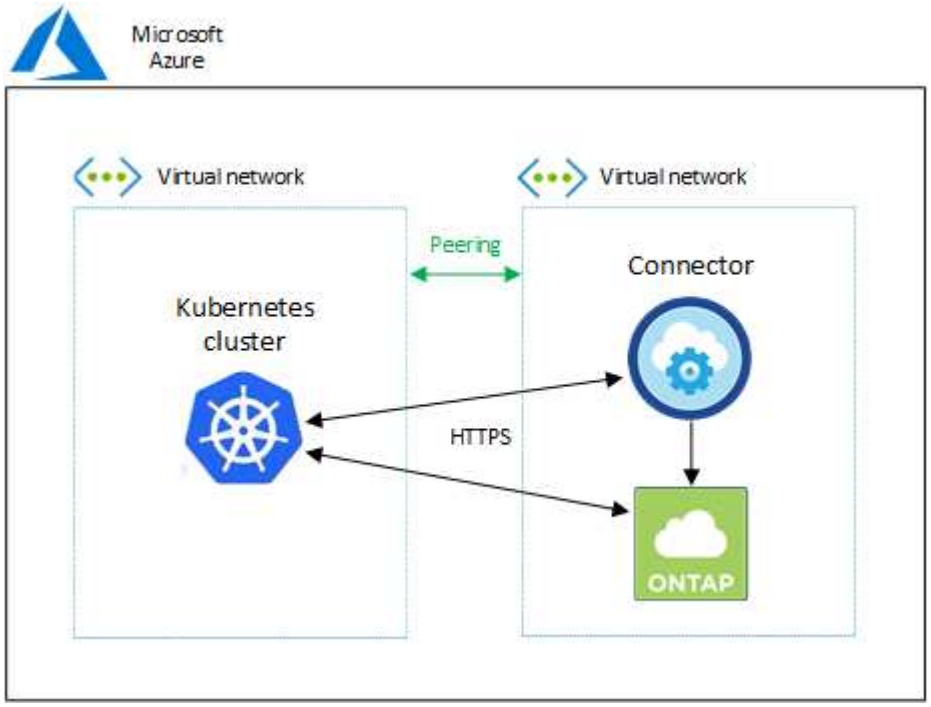
- Jeder Kubernetes-Cluster muss über eine eingehende Verbindung vom Connector verfügen
- Der Connector muss über Port 443 eine ausgehende Verbindung zu jedem Kubernetes-Cluster haben

Die einfachste Möglichkeit, diese Konnektivität bereitzustellen, ist die Implementierung von Connector und Cloud Volumes ONTAP im selben vnet wie der Kubernetes-Cluster. Andernfalls müssen Sie eine Peering-Verbindung zwischen den verschiedenen VNets einrichten.

Hier ein Beispiel, das jede Komponente im selben vnet zeigt.



Ein weiteres Beispiel zeigt einen Kubernetes Cluster, der in einem anderen vnet ausgeführt wird. In diesem Beispiel stellt Peering eine Verbindung zwischen dem vnet für den Kubernetes-Cluster und dem vnet für den Connector und Cloud Volumes ONTAP bereit.



Einrichtung der RBAC-Autorisierung

Die RBAC-Validierung erfolgt nur auf Kubernetes Clustern mit aktiviertem Active Directory (AD). Kubernetes-Cluster ohne AD bestehen die Validierung automatisch.

Sie benötigen für jeden Kubernetes Cluster eine Autorisierung der Connector-Rolle, damit der Connector einen Cluster ermitteln und verwalten kann.

Backup und Restore

Für Backup und Restore ist nur eine Grundautorisierung erforderlich.

Fügen Sie Speicherklassen hinzu

Erweiterte Autorisierung ist erforderlich, um Speicherklassen mithilfe von BlueXP hinzuzufügen und den Cluster auf Änderungen am Backend zu überwachen.

Installieren Sie Astra Trident

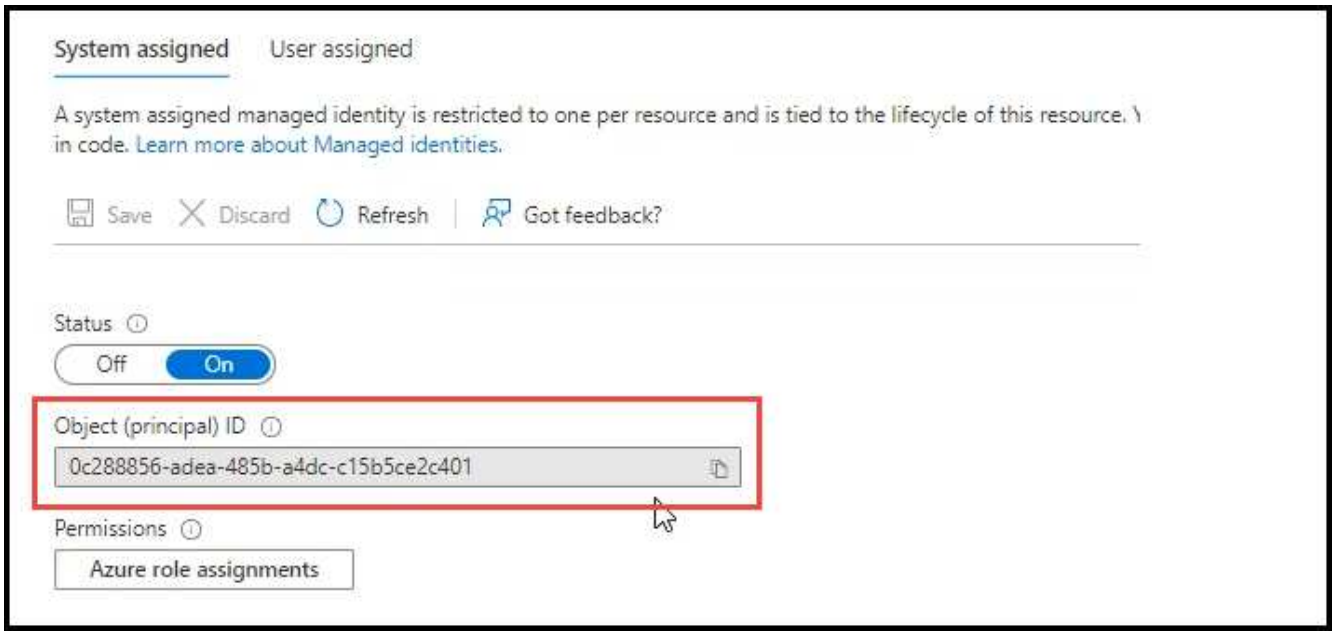
Zur Installation von Astra Trident müssen Sie für BlueXP die vollständige Autorisierung bereitstellen.

Bei der Installation von Astra Trident installiert BlueXP das Astra Trident Back-End und das Kubernetes Secret, das die Zugangsdaten enthält, die Astra Trident zur Kommunikation mit dem Storage-Cluster benötigt.

Bevor Sie beginnen

Ihre RBAC `subjects: name:` Die Konfiguration variiert basierend auf Ihrem Kubernetes-Cluster-Typ leicht.

- Wenn Sie einen **verwalteten AKS-Cluster** bereitstellen, benötigen Sie die Objekt-ID für die vom System zugewiesene verwaltete Identität für den Connector. Diese ID steht im Azure-Managementportal zur Verfügung.



- Wenn Sie ein **selbst verwaltetes Kubernetes Cluster** bereitstellen, benötigen Sie den Benutzernamen eines autorisierten Benutzers.

Schritte

Erstellen Sie eine Cluster-Rolle und Rollenbindung.

1. Erstellen Sie eine YAML-Datei, die den folgenden Text enthält, der auf Ihren Autorisierungsanforderungen basiert. Ersetzen Sie den `subjects: kind:` Variable mit Ihrem Benutzernamen und `subjects: user:` Entweder mit der Objekt-ID für die vom System zugewiesene verwaltete Identität oder mit dem Benutzernamen eines autorisierten Benutzers, wie oben beschrieben.

Backup/Restore

Fügen Sie eine grundlegende Autorisierung hinzu, um Backup und Restore für Kubernetes-Cluster zu ermöglichen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
        - ''
      resources:
        - namespaces
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumes
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - pods
        - pods/exec
      verbs:
        - get
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumeclaims
      verbs:
        - list
        - create
        - watch
    - apiGroups:
        - storage.k8s.io
      resources:
        - storageclasses
      verbs:
```

```
            - list
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentbackends
      verbs:
            - list
            - watch
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentorchestrators
      verbs:
            - get
            - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: User
      name:
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Speicherklassen

Fügen Sie erweiterte Berechtigungen hinzu, um Speicherklassen mithilfe von BlueXP hinzuzufügen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
            - ''
      resources:
            - secrets
            - namespaces
            - persistentvolumeclaims
            - persistentvolumes
            - pods
```

```
            - pods/exec
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
    - apiGroups:
          - storage.k8s.io
      resources:
          - storageclasses
      verbs:
          - get
          - create
          - list
          - watch
          - delete
          - patch
    - apiGroups:
          - trident.netapp.io
      resources:
          - tridentbackends
          - tridentorchestrators
          - tridentbackendconfigs
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: User
      name:
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Installation Von Trident

Über die Befehlszeile erhalten Sie die vollständige Autorisierung, und BlueXP kann Astra Trident installieren.

```
kubectl create clusterrolebinding test --clusterrole cluster-admin
--user <Object (principal) ID>
```

2. Wenden Sie die Konfiguration auf ein Cluster an.

```
kubectl apply -f <file-name>
```

Anforderungen für Kubernetes-Cluster in Google Cloud

Verwaltete Google Kubernetes Engine (GKE)-Cluster und automatisierte Kubernetes-Cluster können in Google mithilfe von BlueXP hinzugefügt und gemanagt werden. Bevor Sie die Cluster zu BlueXP hinzufügen können, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

In diesem Thema wird *Kubernetes Cluster* verwendet, wobei die Konfiguration für GKE und selbst gemanagte Kubernetes Cluster die gleiche ist. Der Cluster-Typ wird bei unterschiedlich der Konfiguration angegeben.

Anforderungen

Astra Trident

Eine der vier aktuellsten Versionen von Astra Trident ist erforderlich. Sie können Astra Trident direkt von BlueXP installieren oder aktualisieren. Sollten Sie "Prüfen Sie die Voraussetzungen" Vor der Installation von Astra Trident

Cloud Volumes ONTAP

Cloud Volumes ONTAP muss sich unter BlueXP im Rahmen desselben Mandanten-Kontos, Arbeitsumgebung und Konnektors befinden wie der Kubernetes-Cluster. "In der Astra Trident Dokumentation finden Sie die Konfigurationsschritte".

BlueXP Connector

Ein Connector muss in Google mit den erforderlichen Berechtigungen ausgeführt werden. Weitere Informationen finden Sie unten.

Netzwerk-Konnektivität

Zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und Cloud Volumes ONTAP ist eine Netzwerkverbindung erforderlich. Weitere Informationen finden Sie unten.

RBAC-Autorisierung

BlueXP unterstützt RBAC-fähige Cluster mit und ohne Active Directory. Die BlueXP Connector-Rolle muss für jedes GKE-Cluster autorisiert sein. Weitere Informationen finden Sie unten.

Bereiten Sie einen Konnektor vor

Für das Erkennen und Managen von Kubernetes-Clustern ist ein BlueXP Connector in Google erforderlich. Sie müssen einen neuen Konnektor erstellen oder einen vorhandenen Konnektor verwenden, der über die erforderlichen Berechtigungen verfügt.

Erstellen Sie einen neuen Konnektor

Folgen Sie den Schritten in einem der nachfolgenden Links.

- "Erstellen Sie einen Connector von BlueXP" (Empfohlen)
- "Installieren Sie den Connector auf einem vorhandenen Linux-Host"

Fügen Sie die erforderlichen Berechtigungen einem vorhandenen Konnektor hinzu (um ein verwaltetes GKE-Cluster zu ermitteln).

Wenn Sie ein verwaltetes GKE-Cluster ermitteln möchten, müssen Sie möglicherweise die benutzerdefinierte Rolle ändern, damit der Connector die Berechtigungen bereitstellen kann.

Schritte

1. In "Cloud Console", Gehen Sie zur Seite **Rollen**.
2. Wählen Sie in der Dropdown-Liste oben auf der Seite das Projekt oder die Organisation aus, das die Rolle enthält, die Sie bearbeiten möchten.
3. Klicken Sie auf eine benutzerdefinierte Rolle.
4. Klicken Sie auf **Rolle bearbeiten**, um die Berechtigungen der Rolle zu aktualisieren.
5. Klicken Sie auf **Berechtigungen hinzufügen**, um der Rolle folgende neue Berechtigungen hinzuzufügen.

   ```
   container.clusters.get
   container.clusters.list
   ```

6. Klicken Sie auf **Aktualisieren**, um die bearbeitete Rolle zu speichern.

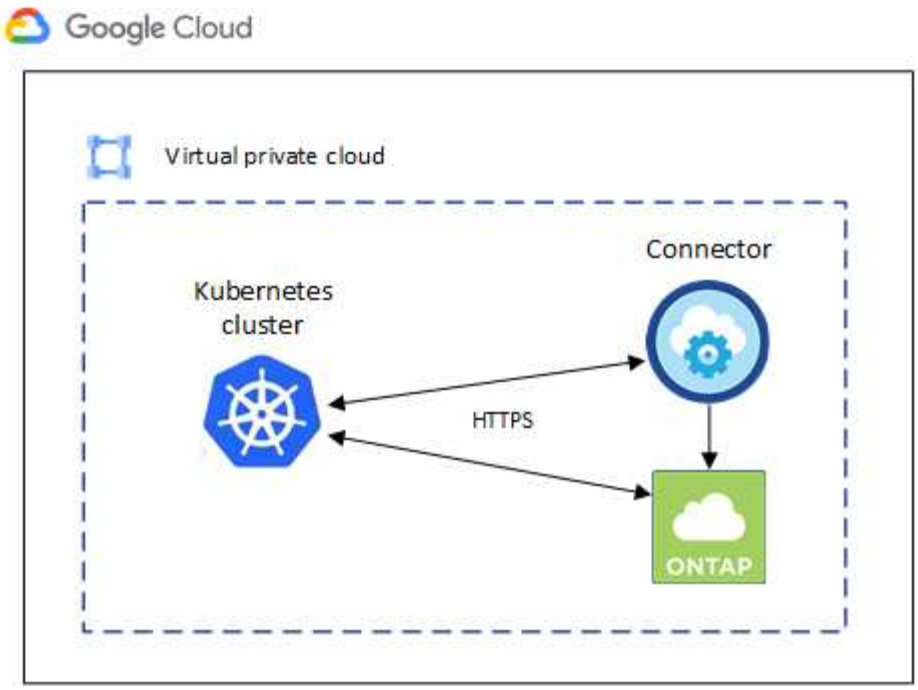
Netzwerkanforderungen prüfen

Sie müssen für die Netzwerkverbindung zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und dem Cloud Volumes ONTAP-System sorgen, das dem Cluster Back-End-Storage bereitstellt.

- Jeder Kubernetes-Cluster muss über eine eingehende Verbindung vom Connector verfügen
- Der Connector muss über Port 443 eine ausgehende Verbindung zu jedem Kubernetes-Cluster haben

Die einfachste Möglichkeit für diese Konnektivität ist die Implementierung von Connector und Cloud Volumes ONTAP in derselben VPC wie der Kubernetes-Cluster. Andernfalls müssen Sie eine Peering-Verbindung zwischen den verschiedenen VPC einrichten.

In diesem Beispiel wird jede Komponente in derselben VPC angezeigt.



Einrichtung der RBAC-Autorisierung

Die RBAC-Validierung erfolgt nur auf Kubernetes Clustern mit aktiviertem Active Directory (AD). Kubernetes-Cluster ohne AD bestehen die Validierung automatisch.

Sie benötigen für jeden Kubernetes Cluster eine Autorisierung der Connector-Rolle, damit der Connector einen Cluster ermitteln und verwalten kann.

Backup und Restore

Für Backup und Restore ist nur eine Grundautorisierung erforderlich.

Fügen Sie Speicherklassen hinzu

Erweiterte Autorisierung ist erforderlich, um Speicherklassen mithilfe von BlueXP hinzuzufügen und den Cluster auf Änderungen am Backend zu überwachen.

Installieren Sie Astra Trident

Zur Installation von Astra Trident müssen Sie für BlueXP die vollständige Autorisierung bereitstellen.

> Bei der Installation von Astra Trident installiert BlueXP das Astra Trident Back-End und das Kubernetes Secret, das die Zugangsdaten enthält, die Astra Trident zur Kommunikation mit dem Storage-Cluster benötigt.

Bevor Sie beginnen

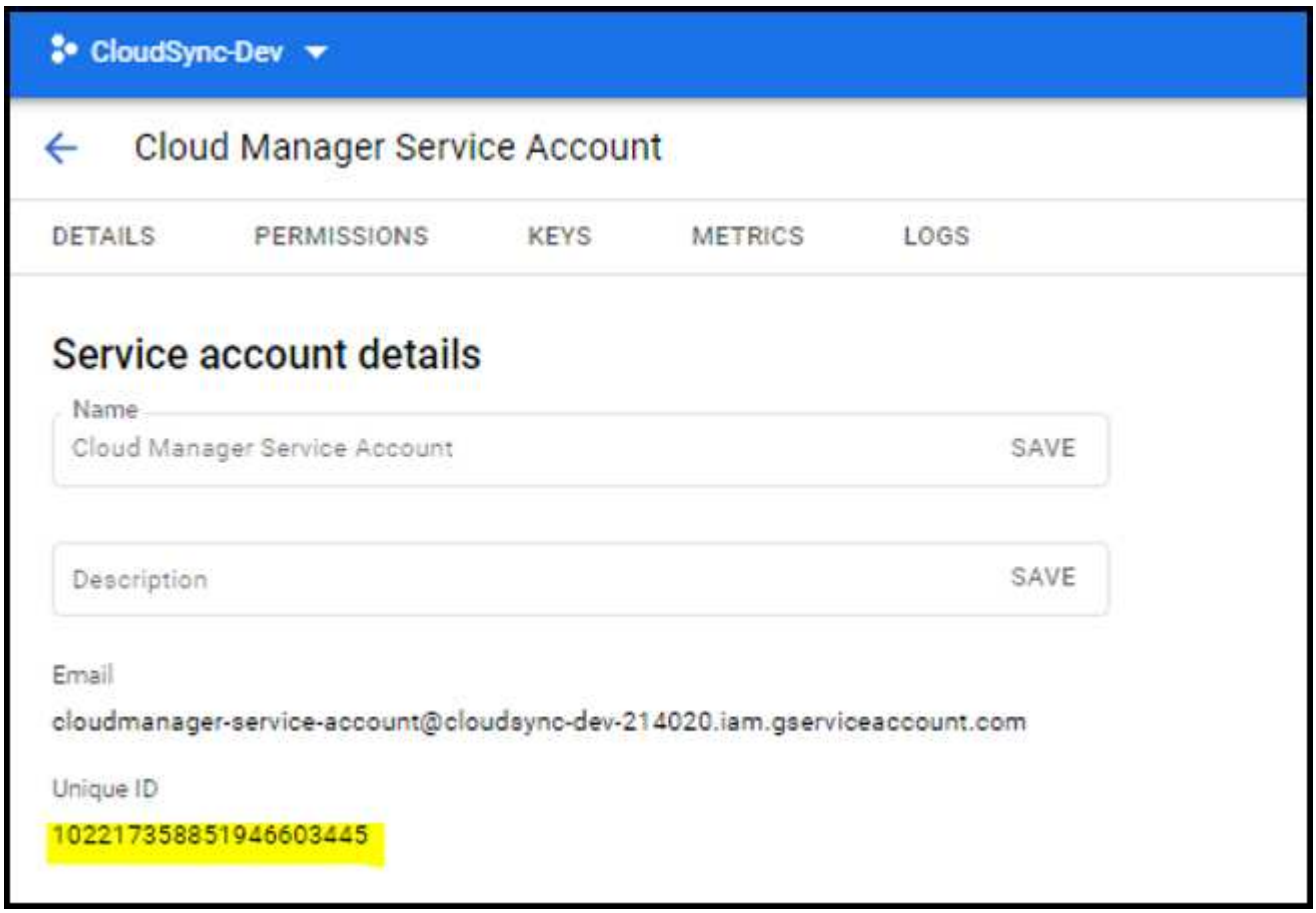
Zu konfigurieren `subjects: name:` In der YAML-Datei müssen Sie die eindeutige BlueXP-ID kennen.

Sie können die eindeutige ID auf zwei Arten finden:

- Verwenden des Befehls:

```
gcloud iam service-accounts list
gcloud iam service-accounts describe <service-account-email>
```

- In den Service-Konto-Details auf dem "Cloud Console".



Schritte

Erstellen Sie eine Cluster-Rolle und Rollenbindung.

1. Erstellen Sie eine YAML-Datei, die den folgenden Text enthält, der auf Ihren Autorisierungsanforderungen basiert. Ersetzen Sie den `subjects: kind:` Variable mit Ihrem Benutzernamen und `subjects: user:` Mit der eindeutigen ID für das autorisierte Servicekonto.

Backup/Restore

Fügen Sie eine grundlegende Autorisierung hinzu, um Backup und Restore für Kubernetes-Cluster zu ermöglichen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
        - ''
      resources:
        - namespaces
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumes
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - pods
        - pods/exec
      verbs:
        - get
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumeclaims
      verbs:
        - list
        - create
        - watch
    - apiGroups:
        - storage.k8s.io
      resources:
        - storageclasses
      verbs:
```

```
            - list
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentbackends
      verbs:
            - list
            - watch
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentorchestrators
      verbs:
            - get
            - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: User
      name:
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Speicherklassen

Fügen Sie erweiterte Berechtigungen hinzu, um Speicherklassen mithilfe von BlueXP hinzuzufügen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
            - ''
      resources:
            - secrets
            - namespaces
            - persistentvolumeclaims
            - persistentvolumes
            - pods
```

```
            - pods/exec
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
    - apiGroups:
          - storage.k8s.io
      resources:
          - storageclasses
      verbs:
          - get
          - create
          - list
          - watch
          - delete
          - patch
    - apiGroups:
          - trident.netapp.io
      resources:
          - tridentbackends
          - tridentorchestrators
          - tridentbackendconfigs
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: User
      name:
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

Installation Von Trident

Über die Befehlszeile erhalten Sie die vollständige Autorisierung, und BlueXP kann Astra Trident installieren.

```
kubectl create clusterrolebinding test --clusterrole cluster-admin
--user <Unique ID>
```

2. Wenden Sie die Konfiguration auf ein Cluster an.

```
kubectl apply -f <file-name>
```

Anforderungen für Kubernetes-Cluster in OpenShift

Selbst gemanagte OpenShift Kubernetes-Cluster können mithilfe von BlueXP hinzugefügt und gemanagt werden. Bevor Sie die Cluster zu BlueXP hinzufügen können, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

Anforderungen

Astra Trident

Eine der vier aktuellsten Versionen von Astra Trident ist erforderlich. Sie können Astra Trident direkt von BlueXP installieren oder aktualisieren. Sollten Sie "Prüfen Sie die Voraussetzungen" Vor der Installation von Astra Trident:

Cloud Volumes ONTAP

Cloud Volumes ONTAP muss als Back-End Storage für den Cluster eingerichtet werden. "In der Astra Trident Dokumentation finden Sie die Konfigurationsschritte".

BlueXP Connector

Für den Import und das Management von Kubernetes-Clustern ist ein BlueXP Connector erforderlich. Sie müssen einen neuen Konnektor erstellen oder einen vorhandenen Konnektor verwenden, der die erforderlichen Berechtigungen für Ihren Cloud-Provider besitzt:

- "AWS Connector"
- "Azure Connector"
- "Google Cloud Connector"

Netzwerk-Konnektivität

Zwischen dem Kubernetes-Cluster und dem Connector sowie zwischen dem Kubernetes-Cluster und Cloud Volumes ONTAP ist eine Netzwerkverbindung erforderlich.

Kubernetes-Konfigurationsdatei (kubeconfig) mit RBAC-Autorisierung

Zum Importieren von OpenShift-Clustern benötigen Sie eine kubeconfig-Datei mit der RBAC-Berechtigung, die erforderlich ist, um verschiedene Funktionen zu ermöglichen. Erstellen Sie eine kubeconfig-Datei.

- Backup und Restore: Backup und Restore erfordern nur grundlegende Autorisierung.

- Hinzufügen von Speicherklassen: Erweiterte Autorisierung ist erforderlich, um Speicherklassen über BlueXP hinzuzufügen und den Cluster auf Änderungen am Backend zu überwachen.
- Installation Astra Trident: Sie müssen über die vollständige Autorisierung für BlueXP verfügen, um Astra Trident zu installieren.

> Bei der Installation von Astra Trident installiert BlueXP das Astra Trident Back-End und das Kubernetes Secret, das die Zugangsdaten enthält, die Astra Trident zur Kommunikation mit dem Storage-Cluster benötigt.

Erstellen Sie eine kubeconfig-Datei

Erstellen Sie mit der OpenShift-CLI eine kubeconfig-Datei für den Import in BlueXP.

Schritte

1. Melden Sie sich über an der OpenShift-CLI an `oc login` Auf eine öffentliche URL mit einem administrativen Benutzer.
2. Erstellen Sie ein Service-Konto wie folgt:
   a. Erstellen Sie eine Dienstkontendatei mit dem Namen `oc-service-account.yaml`.

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

```
oc-service-account.yaml
```

+

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

   a. Wenden Sie das Servicekonto an:

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

3. Erstellen Sie basierend auf Ihren Autorisierungsanforderungen eine benutzerdefinierte Rollenbindung.
   a. Erstellen Sie ein `ClusterRoleBinding` Datei aufgerufen `oc-clusterrolebinding.yaml`.

```
oc-clusterrolebinding.yaml
```

   b. Konfigurieren Sie die RBAC-Autorisierung nach Bedarf für Ihr Cluster.

Backup/Restore

Fügen Sie eine grundlegende Autorisierung hinzu, um Backup und Restore für Kubernetes-Cluster zu ermöglichen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
        - ''
      resources:
        - namespaces
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumes
      verbs:
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - pods
        - pods/exec
      verbs:
        - get
        - list
        - watch
    - apiGroups:
        - ''
      resources:
        - persistentvolumeclaims
      verbs:
        - list
        - create
        - watch
    - apiGroups:
        - storage.k8s.io
      resources:
        - storageclasses
      verbs:
```

```
            - list
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentbackends
      verbs:
            - list
            - watch
    - apiGroups:
            - trident.netapp.io
      resources:
            - tridentorchestrators
      verbs:
            - get
            - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cloudmanager-access-clusterrole
subjects:
    - kind: ServiceAccount
      name: oc-service-account
      namespace: default
```

Speicherklassen

Fügen Sie erweiterte Berechtigungen hinzu, um Speicherklassen mithilfe von BlueXP hinzuzufügen.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
            - ''
      resources:
            - secrets
            - namespaces
            - persistentvolumeclaims
            - persistentvolumes
```

```
          - pods
          - pods/exec
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
    - apiGroups:
          - storage.k8s.io
      resources:
          - storageclasses
      verbs:
          - get
          - create
          - list
          - watch
          - delete
          - patch
    - apiGroups:
          - trident.netapp.io
      resources:
          - tridentbackends
          - tridentorchestrators
          - tridentbackendconfigs
      verbs:
          - get
          - list
          - watch
          - create
          - delete
          - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cloudmanager-access-clusterrole
subjects:
    - kind: ServiceAccount
      name: oc-service-account
      namespace: default
```

Installation Von Trident

Gewähren Sie eine vollständige Administratorautorisierung und aktivieren Sie BlueXP die Installation von Astra Trident.

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

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

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

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

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

Das Ende der Ausgabe sollte wie folgt aussehen:

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

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

5. Erzeugen Sie den kubeconfig wie folgt:

a. Erstellen Sie ein `create-kubeconfig.sh` Datei: Austausch `TOKEN_INDEX` Am Anfang des folgenden Skripts mit dem korrekten Wert.

create-kubeconfig.sh

```
# 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=oc-service-account
NAMESPACE=default
NEW_CONTEXT=oc
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
```

Ergebnis

Sie werden das resultierende verwenden `kubeconfig-sa` Datei zum Hinzufügen eines OpenShift-Clusters zu BlueXP.