



## 要件 Kubernetes clusters

NetApp  
June 23, 2022

# 目次

要件 .....	1
AWS での Kubernetes クラスタの要件 .....	1
Azure での Kubernetes クラスタの要件 .....	10
Google Cloud の Kubernetes クラスタの要件 .....	17

# 要件

## AWS での Kubernetes クラスタの要件

AWS 上の管理対象の Amazon Elastic Kubernetes Service (EKS) クラスタまたは自己管理型の Kubernetes クラスタを Cloud Manager に追加できます。Cloud Manager にクラスタを追加する前に、次の要件を満たしていることを確認する必要があります。



このトピックでは、\_Kubernetes cluster\_where configuration is the same for EKS and selfmanaged Kubernetes clusters を使用します。クラスタタイプは設定が異なる場所で指定します。

### 要件

#### Astra Trident

最新バージョンの 4 つの Astra Trident が必要です。Trident は Cloud Manager から直接インストールできます。お勧めします ["前提条件を確認します"](#) Astra Trident をインストールする前に、

Astra Trident をアップグレードするには、["オペレータにアップグレードしてください"](#)。

#### Cloud Volumes ONTAP

Cloud Volumes ONTAP for AWS は、クラスタのバックエンドストレージとしてセットアップする必要があります。["設定手順については、Astra Trident のドキュメントを参照してください"](#)。

#### Cloud Manager Connector の略

必要な権限を持つコネクタが AWS で実行されている必要があります。[詳細は以下をご覧ください](#)。

#### ネットワーク接続

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタと Cloud Volumes ONTAP の間にはネットワーク接続が必要です。[詳細は以下をご覧ください](#)。

#### RBAC 許可

Cloud Manager Connector ロールは、各 Kubernetes クラスタで許可されている必要があります。[詳細は以下をご覧ください](#)。

### コネクタを準備します

Kubernetes クラスタを検出および管理するには、AWS で Cloud Manager Connector を使用する必要があります。新しいコネクタを作成するか、必要な権限を持つ既存のコネクタを使用する必要があります。

#### 新しいコネクタを作成します

次のリンクのいずれかの手順に従います。

- ["Cloud Manager からコネクタを作成します"](#) (推奨)
- ["AWS Marketplace からコネクタを作成します"](#)

- "AWS の既存の Linux ホストにコネクタをインストールします"

必要な権限を既存のコネクタに追加します

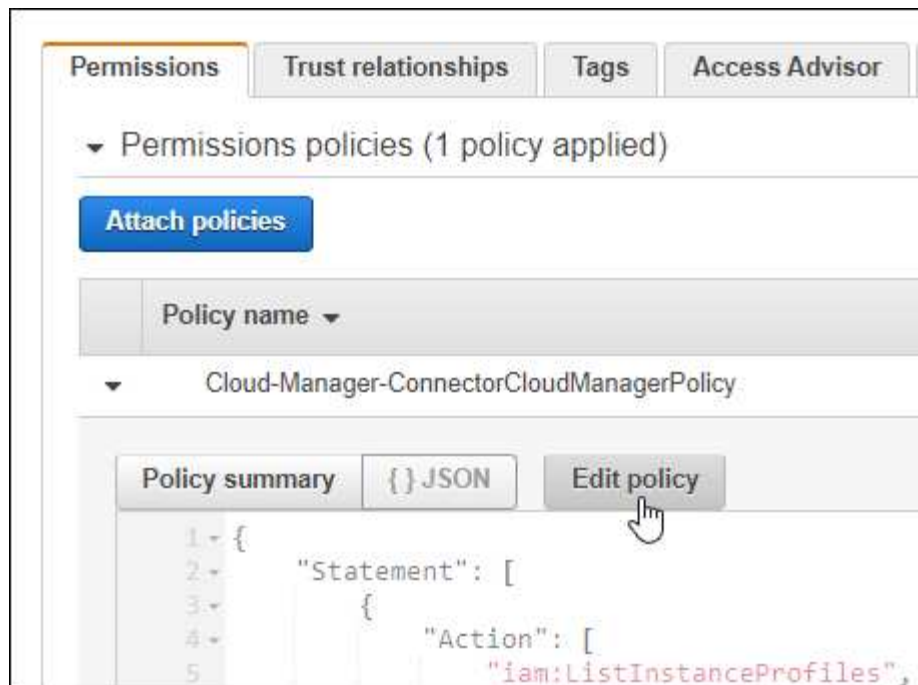
3.9.13 リリース以降、new\_newly で作成されたコネクタには、Kubernetes クラスタの検出と管理を可能にする新しい AWS 権限が 3 つ含まれています。このリリースよりも前のリリースでコネクタを作成していた場合は、権限を付与するために、コネクタの IAM ロールの既存のポリシーを変更する必要があります。

手順

1. AWS コンソールにアクセスして EC2 サービスを開きます。
2. コネクタインスタンスを選択し、\*セキュリティ\* をクリックして、IAM ロールの名前をクリックし、IAM サービスでロールを表示します。



3. [\* アクセス許可 \*] タブで、ポリシーを展開し、[\* ポリシーの編集 \*] をクリックします。



4. JSON \* をクリックして、最初のアクションセットに次の権限を追加します。

- EC2: DescribeRegions (説明領域)
- EKS : リストクラスタ
- EKS : DescribeCluster
- IAM : GetInstanceProfile

"ポリシーの完全な JSON 形式を表示します"。

5. [ ポリシーの確認 ] をクリックし、[ 変更の保存 ] をクリックします。

## ネットワーク要件を確認します

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタとクラスタにバックエンドストレージを提供する Cloud Volumes ONTAP システムとの間にネットワーク接続を提供する必要があります。

- 各 Kubernetes クラスタがコネクタからインバウンド接続を確立している必要があります
- コネクタには、ポート 443 経由で各 Kubernetes クラスタへのアウトバウンド接続が必要です

この接続を確立する最も簡単な方法は、Kubernetes クラスタと同じ VPC にコネクタと Cloud Volumes ONTAP を導入することです。VPC が確立されていない場合は、VPC 間に VPC ピアリング接続を設定する必要があります。

以下は、同じ VPC 内の各コンポーネントの例です。



別の VPC で実行されている EKS クラスターを次に示します。この例では、VPC ピアリングによって、EKS クラスターの VPC とコネクタおよび Cloud Volumes ONTAP の VPC 間の接続が確立されます。



## RBAC 許可をセットアップします

コネクタがクラスターを検出して管理できるように、各 Kubernetes クラスターで Connector ロールを承認する必要があります。

異なる機能を有効にするには、異なる許可が必要です。

### バックアップとリストア

バックアップとリストアに必要なのは、基本的な許可だけです。

## ストレージクラスを追加する

Cloud Manager を使用してストレージクラスを追加するには、拡張された許可が必要です。

### **Astra Trident** をインストールします

Cloud Manager が Astra Trident をインストールするための完全な権限を付与する必要があります。



Astra Trident をインストールすると、Cloud Manager は Astra Trident バックエンドと、Astra Trident がストレージクラスと通信するために必要なクレデンシャルを含む Kubernetes シークレットをインストールします。

## 手順

1. クラスターロールとロールバインドを作成します。
  - a. 許可要件に基づいて次のテキストを含む YAML ファイルを作成します。

## バックアップ/リストア

Kubernetes クラスタのバックアップとリストアを有効にするための基本的な許可を追加する。

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



```

    verbs:
      - list
  - apiGroups:
      - trident.netapp.io
    resources:
      - tridentorchestrators
    verbs:
      - get
---
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

```

## ストレージクラス

拡張された権限を追加し、Cloud Manager を使用してストレージクラスを追加します。

```

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
      - create
      - delete
  - apiGroups:

```

```

      - storage.k8s.io
resources:
  - storageclasses
verbs:
  - get
  - create
  - list
  - delete
  - patch
- apiGroups:
  - trident.netapp.io
resources:
  - tridentbackends
  - tridentorchestrators
  - tridentbackendconfigs
verbs:
  - get
  - list
  - create
  - delete

---
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

```

### Trident をインストール

コマンドラインを使用して完全な権限を付与し、Cloud Manager が Astra Trident をインストールできるようにします。

```

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

```

b. クラスタに構成を適用します。

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

## 2. 権限グループへの ID マッピングを作成します。

**eksctl** を使用します

eksctl を使用して、クラスタと Cloud Manager Connector 用の IAM ロールの間に IAM ID マッピングを作成します。

["eksctl のマニュアルを参照してください"](#)。

以下に例を示します。

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

**aws-auth** を編集します

AWS- auth ConfigMap を直接編集して、Cloud Manager Connector の IAM ロールに RBAC アクセスを追加します。

["詳細な手順については、AWS EKS のドキュメントを参照してください"](#)。

以下に例を示します。

```
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
```

# Azure での Kubernetes クラスタの要件

Cloud Manager を使用して、Azure で管理対象 Azure Kubernetes クラスタ（AKS）と自己管理型 Kubernetes クラスタを追加および管理できます。Cloud Manager にクラスタを追加する前に、次の要件を満たしていることを確認してください。



このトピックでは、\_Kubernetes cluster\_where configuration is the same for AKS and selfmanaged Kubernetes clusters を使用します。クラスタタイプは設定が異なる場所で指定します。

## 要件

### Astra Trident

最新バージョンの 4 つの Astra Trident が必要です。Trident は Cloud Manager から直接インストールできます。お勧めします ["前提条件を確認します"](#) Astra Trident をインストールする前に、

Astra Trident をアップグレードするには、["オペレータにアップグレードしてください"](#)。

### Cloud Volumes ONTAP

クラスタのバックエンドストレージとして Cloud Volumes ONTAP が設定されている必要があります。["設定手順については、Astra Trident のドキュメントを参照してください"](#)。

### Cloud Manager Connector の略

必要な権限を持つコネクタが Azure で実行されている必要があります。[詳細は以下をご覧ください](#)。

### ネットワーク接続

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタと Cloud Volumes ONTAP の間にはネットワーク接続が必要です。[詳細は以下をご覧ください](#)。

### RBAC 許可

Cloud Manager は、Active Directory を使用するかどうかに関係なく、RBAC 対応のクラスタをサポートします。Cloud Manager Connector ロールは、各 Azure クラスタで許可されている必要があります。[詳細は以下をご覧ください](#)。

## コネクタを準備します

Kubernetes クラスタを検出および管理するには、Azure で Cloud Manager Connector を使用する必要があります。新しいコネクタを作成するか、必要な権限を持つ既存のコネクタを使用する必要があります。

### 新しいコネクタを作成します

次のリンクのいずれかの手順に従います。

- ["Cloud Manager からコネクタを作成します"](#)（推奨）
- ["Azure Marketplace からコネクタを作成します"](#)
- ["既存の Linux ホストにコネクタをインストールします"](#)

既存のコネクタに必要な権限を追加する（管理対象の **AKS** クラスタを検出する）

管理対象の AKS クラスタを検出するには、コネクタのカスタムロールを変更して権限を提供しなければならない場合があります。

#### 手順

1. Connector 仮想マシンに割り当てられているロールを特定します。
  - a. Azure ポータルで、仮想マシンサービスを開きます。
  - b. Connector 仮想マシンを選択します。
  - c. [ 設定 ] で、[ **Identity** ] を選択します。
  - d. Azure の役割の割り当て \* をクリックします。
  - e. Connector 仮想マシンに割り当てられているカスタムロールをメモしておきます。
2. カスタムロールを更新します。
  - a. Azure ポータルで、Azure サブスクリプションを開きます。
  - b. [ \* アクセス制御 ( IAM ) ] > [ 役割 \* ] をクリックします。
  - c. カスタムロールの省略記号 (...) をクリックし、 \* 編集 \* をクリックします。
  - d. JSON をクリックして、次の権限を追加します。

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

- e. [ \* Review + update \* ] をクリックし、[ \* Update \* ] をクリックします。

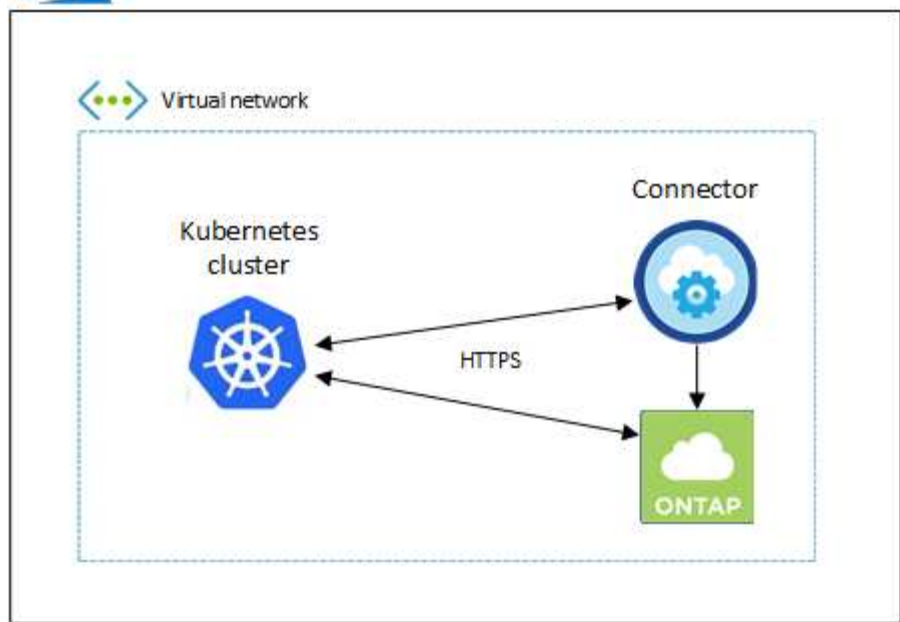
## ネットワーク要件を確認します

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタとクラスタにバックエンドストレージを提供する Cloud Volumes ONTAP システムとの間にネットワーク接続を提供する必要があります。

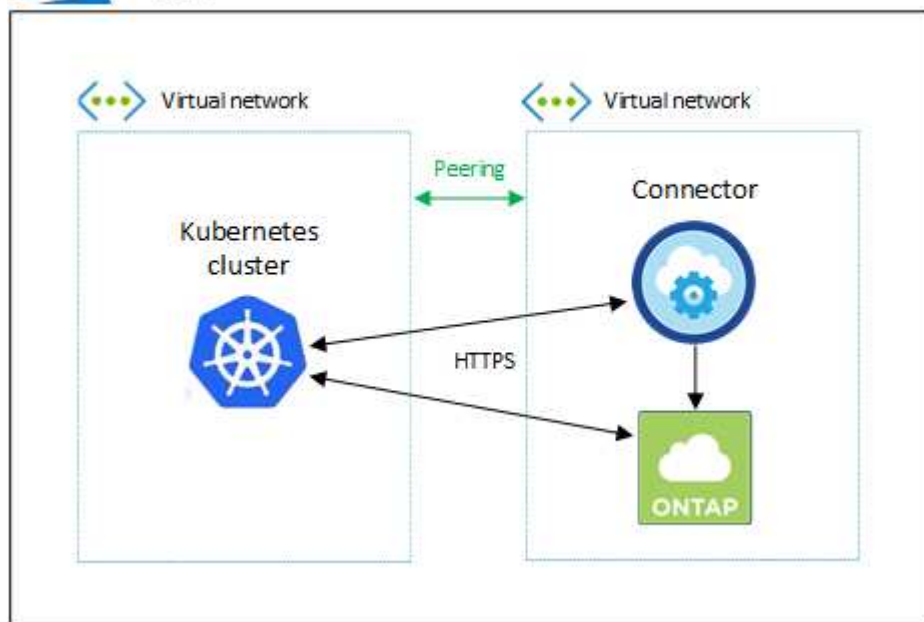
- 各 Kubernetes クラスタがコネクタからインバウンド接続を確立している必要があります
- コネクタには、ポート 443 経由で各 Kubernetes クラスタへのアウトバウンド接続が必要です

この接続を確立する最も簡単な方法は、Kubernetes クラスタと同じ VNet にコネクタと Cloud Volumes ONTAP を導入することです。それ以外の場合は、異なる VNet 間のピアリング接続を設定する必要があります。

以下は、同じ VNet 内の各コンポーネントの例です。



別の VNet で実行される Kubernetes クラスタの例を次に示します。この例では、ピアリングによって Kubernetes クラスタの VNet とコネクタおよび Cloud Volumes ONTAP の VNet 間の接続が確立されます。



## RBAC 許可をセットアップします

RBAC の検証は、Active Directory (AD) が有効になっている Kubernetes クラスタでのみ実行されます。AD を使用しない Kubernetes クラスタは、検証に自動的に合格します。

コネクタがクラスタを検出して管理できるように、各 Kubernetes クラスタで Connector ロールを承認する必要があります。

## バックアップとリストア

バックアップとリストアに必要なのは、基本的な許可だけです。

## ストレージクラスを追加する

Cloud Manager を使用してストレージクラスを追加するには、拡張された許可が必要です。

## Astra Trident をインストールします

Cloud Manager が Astra Trident をインストールするための完全な権限を付与する必要があります。



Astra Trident をインストールすると、Cloud Manager は Astra Trident バックエンドと、Astra Trident がストレージクラスと通信するために必要なクレデンシャルを含む Kubernetes シークレットをインストールします。

RBAC のサブジェクト名 : name:` の構成は、Kubernetes クラスタのタイプによって若干異なります。

- 管理対象 AKS クラスタ \* を導入する場合、コネクタにシステムが割り当てた管理 ID のオブジェクト ID が必要です。この ID は Azure 管理ポータルで入手できます。

The screenshot shows the 'System assigned' tab in the Azure portal. It includes a description of system assigned managed identities, action buttons (Save, Discard, Refresh, Got feedback?), a 'Status' toggle set to 'On', and an 'Object (principal) ID' field containing '0c288856-adea-485b-a4dc-c15b5ce2c401'. A red box highlights the ID field. Below is a 'Permissions' section with an 'Azure role assignments' button.

- 自己管理型の Kubernetes クラスタ \* を導入する場合は、許可されたユーザのユーザ名が必要です。

クラスタロールとロールバインドを作成します。

1. 許可要件に基づいて次のテキストを含む YAML ファイルを作成します。「Subjects:kind」変数をユーザ名に置き換え、「Subjects:user:`」をシステムに割り当てられた管理対象 ID のオブジェクト ID または上記の権限を持つユーザのユーザ名に置き換えます。

## バックアップ/リストア

Kubernetes クラスタのバックアップとリストアを有効にするための基本的な許可を追加する。

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



```

    verbs:
      - list
  - apiGroups:
      - trident.netapp.io
    resources:
      - tridentorchestrators
    verbs:
      - get
---
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

```

## ストレージクラス

拡張された権限を追加し、Cloud Manager を使用してストレージクラスを追加します。

```

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
      - create
      - delete
  - apiGroups:

```

```

      - storage.k8s.io
resources:
  - storageclasses
verbs:
  - get
  - create
  - list
  - delete
  - patch
- apiGroups:
  - trident.netapp.io
resources:
  - tridentbackends
  - tridentorchestrators
  - tridentbackendconfigs
verbs:
  - get
  - list
  - create
  - delete
---
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

```

### Trident をインストール

コマンドラインを使用して完全な権限を付与し、Cloud Manager が Astra Trident をインストールできるようにします。

```

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

```

## 2. クラスタに構成を適用します。

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

## Google Cloud の Kubernetes クラスタの要件

Cloud Manager を使用して、Google で管理対象 Google Kubernetes Engine (GKE) クラスタと自己管理型 Kubernetes クラスタを追加および管理できます。Cloud Manager にクラスタを追加する前に、次の要件を満たしていることを確認してください。



このトピックでは、\_Kubernetes cluster\_where 構成は、GKE クラスタと自己管理型 Kubernetes クラスタで同じです。クラスタタイプは設定が異なる場所で指定します。

### 要件

#### Astra Trident

最新バージョンの 4 つの Astra Trident が必要です。Trident は Cloud Manager から直接インストールできます。お勧めします ["前提条件を確認します"](#) Astra Trident をインストールする前に

Astra Trident をアップグレードするには、["オペレータにアップグレードしてください"](#)。

#### Cloud Volumes ONTAP

Cloud Volumes ONTAP は、Kubernetes クラスタと同じテナンシーアカウント、ワークスペース、コネクタで Cloud Manager に配置する必要があります。["設定手順については、Astra Trident のドキュメントを参照してください"](#)。

#### Cloud Manager Connector の略

必要な権限を持つ Connector が Google で実行されている必要があります。[詳細は以下をご覧ください](#)。

#### ネットワーク接続

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタと Cloud Volumes ONTAP の間にはネットワーク接続が必要です。[詳細は以下をご覧ください](#)。

#### RBAC 許可

Cloud Manager は、Active Directory を使用するかどうかに関係なく、RBAC 対応のクラスタをサポートします。Cloud Manager Connector ロールは、各 GKE クラスタで許可されている必要があります。[詳細は以下をご覧ください](#)。

### コネクタを準備します

Kubernetes クラスタを検出および管理するには、Google の Cloud Manager Connector が必要です。新しいコネクタを作成するか、必要な権限を持つ既存のコネクタを使用する必要があります。

#### 新しいコネクタを作成します

次のリンクのいずれかの手順に従います。

- ["Cloud Manager からコネクタを作成します"](#) (推奨)

- "既存の Linux ホストにコネクタをインストールします"

既存のコネクタに必要な権限を追加する（管理対象の **GKE** クラスタを検出するため）

管理対象 GKE クラスタを検出する場合は、コネクタのカスタムロールを変更して権限を付与する必要があります。

手順

1. インチ "[Cloud Console の略](#)"をクリックし、\* Roles \* ページに移動します。
2. ページ上部のドロップダウンリストを使用して、編集するロールを含むプロジェクトまたは組織を選択します。
3. カスタムロールをクリックします。
4. 役割の権限を更新するには、\* 役割の編集 \* をクリックします。
5. [ 権限の追加 \*] をクリックして、次の新しい権限を役割に追加します。

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

6. [ 更新（Update） ] をクリックして、編集したロールを保存する。

## ネットワーク要件を確認します

Kubernetes クラスタとコネクタの間、および Kubernetes クラスタとクラスタにバックエンドストレージを提供する Cloud Volumes ONTAP システムとの間にネットワーク接続を提供する必要があります。

- 各 Kubernetes クラスタがコネクタからインバウンド接続を確立している必要があります
- コネクタには、ポート 443 経由で各 Kubernetes クラスタへのアウトバウンド接続が必要です

この接続を確立する最も簡単な方法は、Kubernetes クラスタと同じ VPC にコネクタと Cloud Volumes ONTAP を導入することです。それ以外の場合は、異なる VPC 間にピア接続を設定する必要があります。

以下は、同じ VPC 内の各コンポーネントの例です。



## RBAC 許可をセットアップします

RBAC の検証は、Active Directory（AD）が有効になっている Kubernetes クラスタでのみ実行されます。AD を使用しない Kubernetes クラスタは、検証に自動的に合格します。

コネクタがクラスタを検出して管理できるように、各 Kubernetes クラスタで Connector ロールを承認する必要があります。

### バックアップとリストア

バックアップとリストアに必要なのは、基本的な許可だけです。

### ストレージクラスを追加する

Cloud Manager を使用してストレージクラスを追加するには、拡張された許可が必要です。

## Astra Trident をインストールします

Cloud Manager が Astra Trident をインストールするための完全な権限を付与する必要があります。



Astra Trident をインストールすると、Cloud Manager は Astra Trident バックエンドと、Astra Trident がストレージクラスと通信するために必要なクレデンシャルを含む Kubernetes シークレットをインストールします。

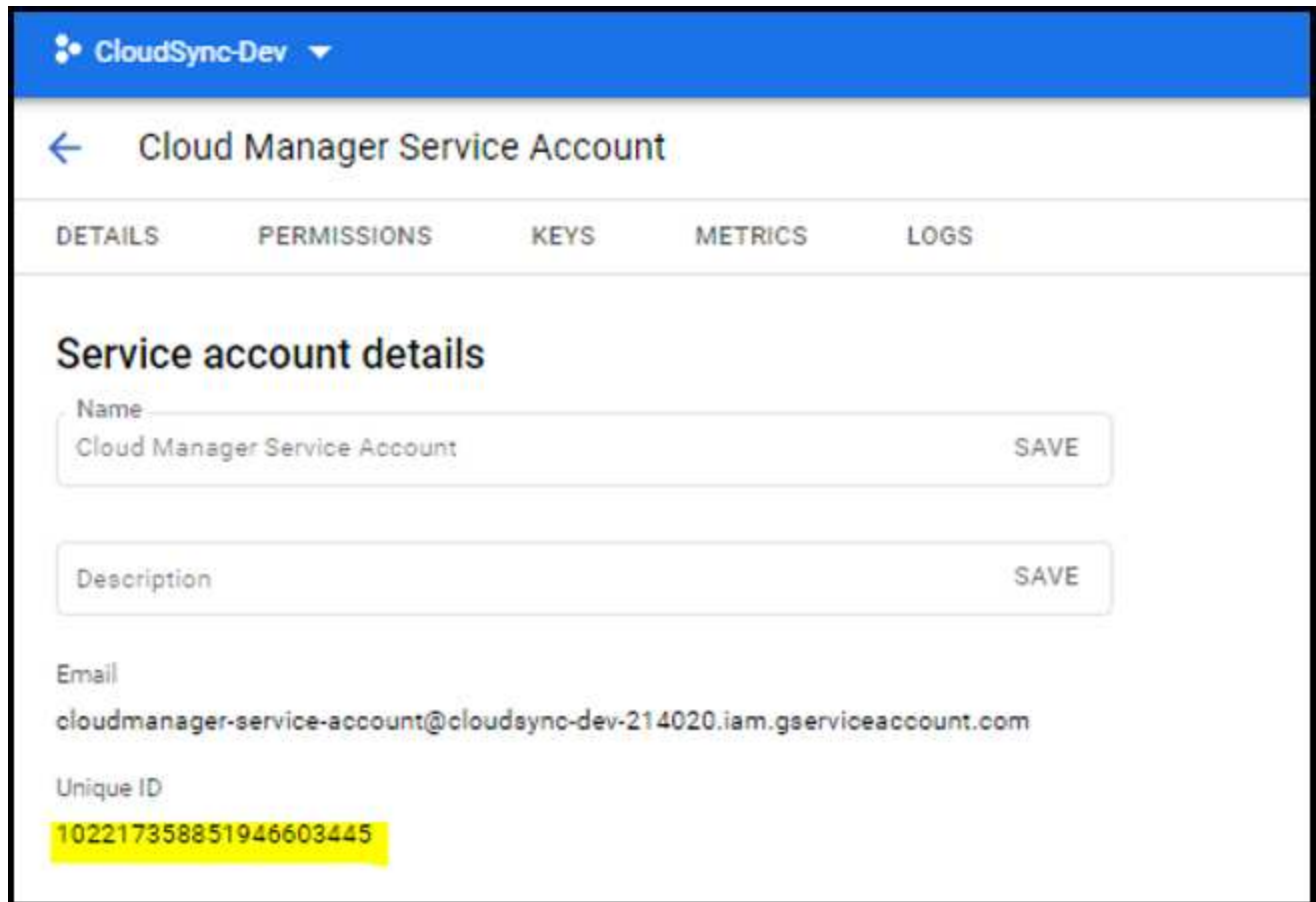
YAML ファイルで「Subjects:name:`」を設定するには、Cloud Manager の一意の ID を知っている必要があります。

一意の ID は、次の 2 つの方法のいずれかで確認できます。

- コマンドを使用します。

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

- のサービスアカウントの詳細で確認します "[Cloud Console の略](#)".



クラスターロールとロールバインドを作成します。

1. 許可要件に基づいて次のテキストを含む YAML ファイルを作成します。「Subjects:kind」変数をユーザー名に置き換え、「Subjects:user:」を認証されたサービスアカウントの一意の ID に置き換えます。

## バックアップ/リストア

Kubernetes クラスタのバックアップとリストアを有効にするための基本的な許可を追加する。

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

```

    verbs:
      - list
  - apiGroups:
      - trident.netapp.io
    resources:
      - tridentorchestrators
    verbs:
      - get
---
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

```

## ストレージクラス

拡張された権限を追加し、Cloud Manager を使用してストレージクラスを追加します。

```

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
      - create
      - delete
  - apiGroups:

```



```

      - storage.k8s.io
resources:
  - storageclasses
verbs:
  - get
  - create
  - list
  - delete
  - patch
- apiGroups:
  - trident.netapp.io
resources:
  - tridentbackends
  - tridentorchestrators
  - tridentbackendconfigs
verbs:
  - get
  - list
  - create
  - delete
---
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

```

### Trident をインストール

コマンドラインを使用して完全な権限を付与し、Cloud Manager が Astra Trident をインストールできるようにします。

```

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

```

## 2. クラスタに構成を適用します。

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

## 著作権情報

Copyright © 2022 NetApp, Inc. All rights reserved. 米国で印刷されていますこのドキュメントは著作権によって保護されています。画像媒体、電子媒体、および写真複写、記録媒体などの機械媒体など、いかなる形式および方法による複製も禁止します。テープ媒体、または電子検索システムへの保管-著作権所有者の書面による事前承諾なし。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、いかなる場合でも、間接的、偶発的、特別、懲罰的、またはまたは結果的損害（代替品または代替サービスの調達、使用の損失、データ、利益、またはこれらに限定されないものを含みますが、これらに限定されません。）ただし、契約、厳格責任、または本ソフトウェアの使用に起因する不法行為（過失やその他を含む）のいずれであっても、かかる損害の可能性について知らされていた場合でも、責任の理論に基づいて発生します。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、またはその他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1 つ以上の米国特許、その他の国の特許、および出願中の特許により特許、その他の国の特許、および出願中の特許。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7103（1988 年 10 月）および FAR 52-227-19（1987 年 6 月）の Rights in Technical Data and Computer Software（技術データおよびコンピュータソフトウェアに関する諸権利）条項の（c）（1）（ii）項、に規定された制限が適用されます。

## 商標情報

NetApp、NetAppのロゴ、に記載されているマーク <http://www.netapp.com/TM> は、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。