



## 要件 Kubernetes clusters

NetApp  
April 04, 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 .....	9
Google Cloud の Kubernetes クラスタの要件 .....	16

# 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。

```
"eks:ListClusters",  
"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. eksctl' を使用して ' アクセス権グループへの ID マッピングを作成します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}}
```

"eksctl のマニュアルを参照してください"。

## 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

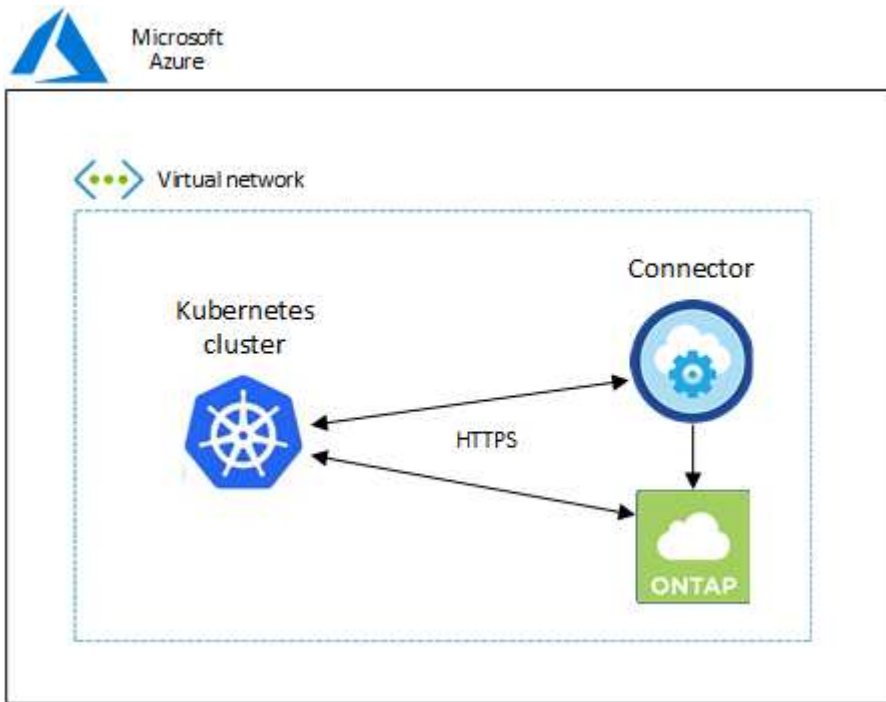
Kubernetes クラスタとコネクタの間、および Kubernetes クラスタとクラスタにバックエンドストレージを提供する Cloud Volumes ONTAP システムとの間にネットワーク接続を提供する必要があります。

- 各 Kubernetes クラスタがコネクタからインバウンド接続を確立する必要があります

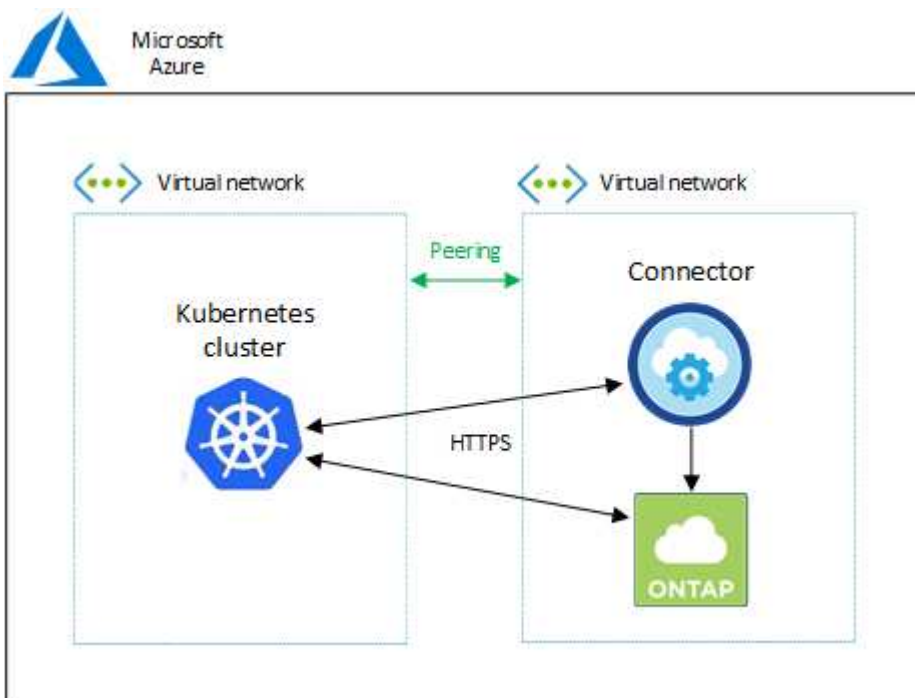
- コネクタには、ポート 443 経由で各 Kubernetes クラスタへのアウトバウンド接続が必要です

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タ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 Azure portal interface for a managed identity. The 'System assigned' tab is selected. A note states: 'A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. \ in code. [Learn more about Managed identities.](#)' Below this are buttons for 'Save', 'Discard', 'Refresh', and 'Got feedback?'. The 'Status' is set to 'On'. The 'Object (principal) ID' field is highlighted with a red box and contains the value '0c288856-adea-485b-a4dc-c15b5ce2c401'. Below it, the 'Permissions' section shows 'Azure role assignments'.

- 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。

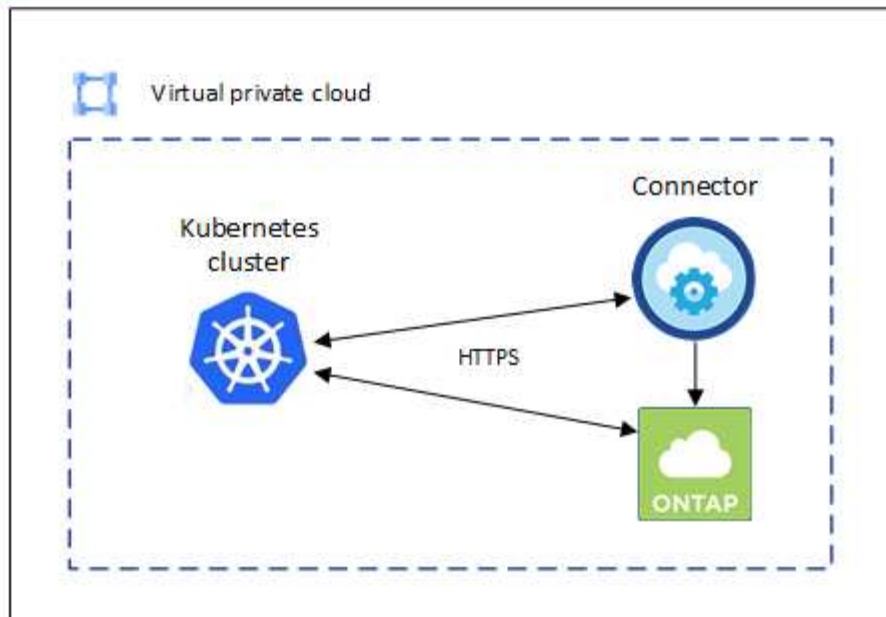
## ネットワーク要件を確認します

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。

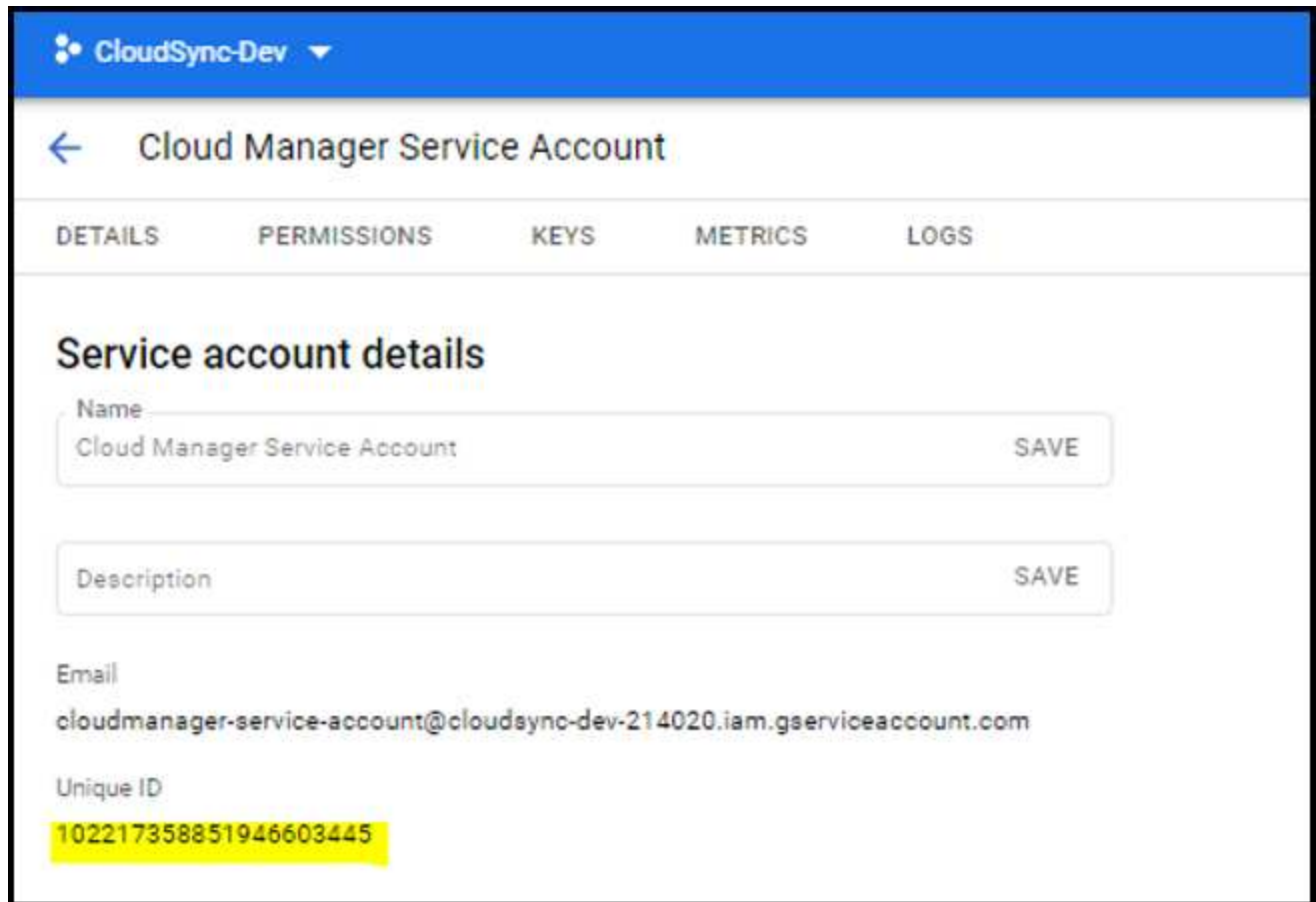
YAML ファイルで「Subjects:name:`」を設定するには、Cloud Manager の一意の ID を知っている必要があります。

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。

```
NEED FROM 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>
```

## Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.