



要求 Kubernetes clusters

NetApp
April 04, 2022

目录

- 要求 1
 - AWS 中 Kubernetes 集群的要求 1
 - Azure 中 Kubernetes 集群的要求 9
 - Google Cloud 中的 Kubernetes 集群的要求 16

要求

AWS 中 Kubernetes 集群的要求

您可以将 AWS 上的受管 Amazon Elastic Kubernetes Service （EKS）集群或自管 Kubernetes 集群添加到 Cloud Manager。在将集群添加到 Cloud Manager 之前，您需要确保满足以下要求。

本主题使用 *Kubernetes cluster*，其中 EKS 和自管理 Kubernetes 集群的配置相同。在配置不同的位置指定集群类型。

要求

Astra Trident

需要使用四个最新版本的 Astra Trident 之一。您可以直接从 Cloud Manager 安装 Astra Trident。您应该 ["查看前提条件"](#) 安装 Astra Trident 之前。

要升级 Astra Trident，["使用操作员升级"](#)。

Cloud Volumes ONTAP

Cloud Volumes ONTAP for AWS 必须设置为集群的后端存储。["有关配置步骤，请转至 Astra Trident 文档"](#)。

Cloud Manager Connector

必须使用所需权限在 AWS 中运行 Connector。[在下方了解更多信息。](#)

网络连接

Kubernetes 集群和 Connector 之间以及 Kubernetes 集群和 Cloud Volumes ONTAP 之间需要网络连接。[在下方了解更多信息。](#)

RBAC 授权

必须在每个 Kubernetes 集群上授权 Cloud Manager Connector 角色。[在下方了解更多信息。](#)

准备连接器

要发现和管理 Kubernetes 集群，AWS 需要使用 Cloud Manager Connector。您需要创建新的 Connector 或使用具有所需权限的现有 Connector。

创建新的 Connector

按照以下链接之一中的步骤进行操作。

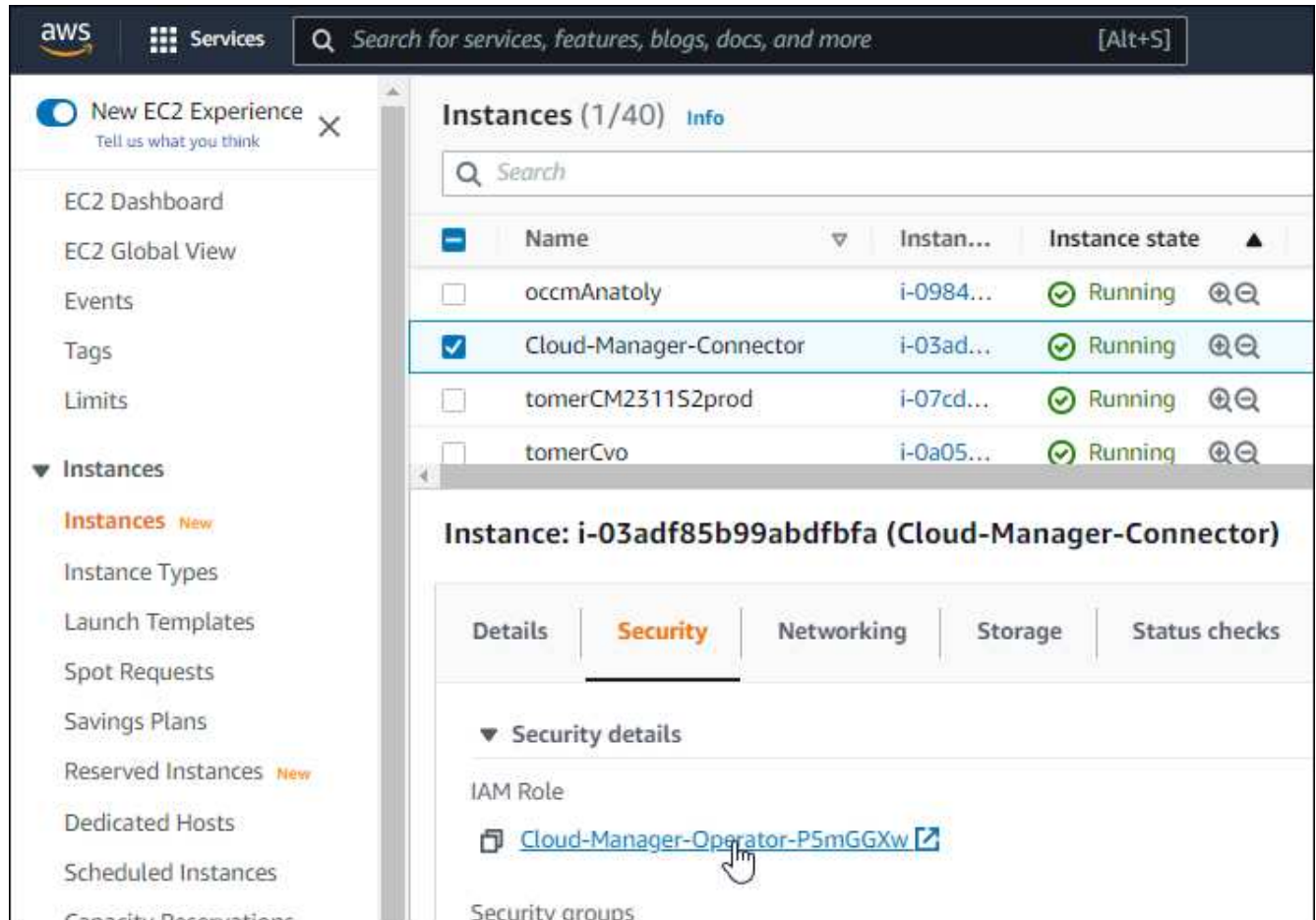
- ["从 Cloud Manager 创建 Connector"](#) 建议
- ["从 AWS Marketplace 创建 Connector"](#)
- ["在 AWS 的现有 Linux 主机上安装 Connector"](#)

将所需权限添加到现有 Connector

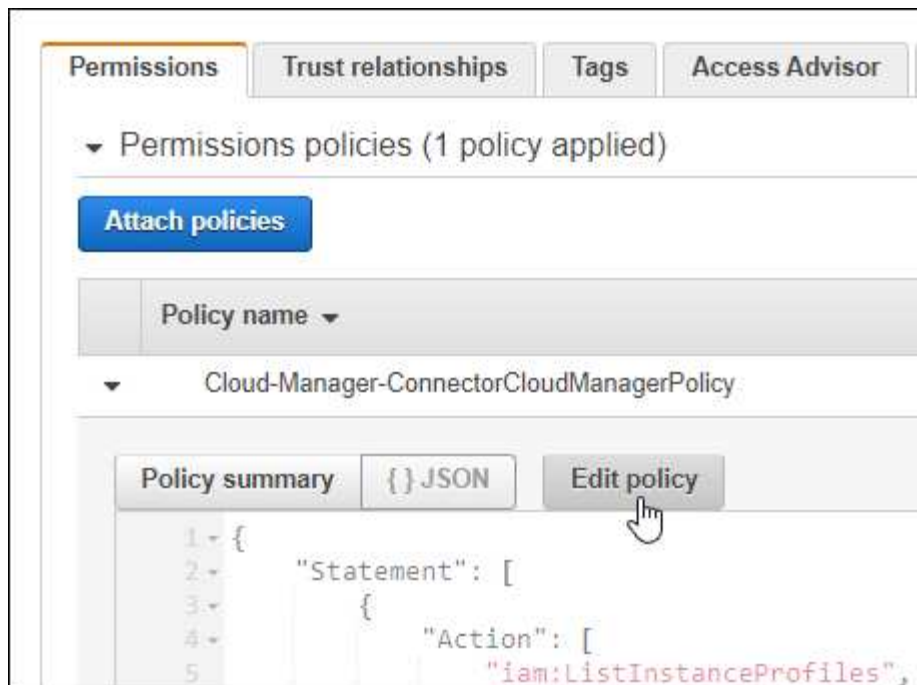
从 3.9.13 版开始，任何 *new* 创建的 Connectors 均包含三个新的 AWS 权限，用于发现和管理 Kubernetes 集群。如果您在此版本之前创建了 Connector，则需要修改此 Connector 的 IAM 角色的现有策略以提供权限。

步骤

1. 转至 AWS 控制台并打开 EC2 服务。
2. 选择 Connector 实例，单击 * 安全性 *，然后单击 IAM 角色的名称以查看 IAM 服务中的角色。



3. 在 * 权限 * 选项卡中，展开策略并单击 * 编辑策略 *。



4. 单击 *。JSON*，然后在第一组操作下添加以下权限：

```
"eks:ListClusters",  
"eks:DescribeCluster",  
"iam:GetInstanceProfile"
```

"查看策略的完整 JSON 格式"。

5. 单击 * 查看策略 *，然后单击 * 保存更改 *。

查看网络连接要求

您需要在 Kubernetes 集群和 Connector 之间以及 Kubernetes 集群与为集群提供后端存储的 Cloud Volumes ONTAP 系统之间提供网络连接。

- 每个 Kubernetes 集群都必须与 Connector 建立入站连接
- 此连接器必须通过端口 443 与每个 Kubernetes 集群建立出站连接

提供此连接的最简单方法是，将连接器和 Cloud Volumes ONTAP 部署在与 Kubernetes 集群相同的 VPC 中。否则，您需要在不同的 VPC 之间设置 VPC 对等连接。

以下示例显示了同一 VPC 中的每个组件。



下面是另一个示例，显示了一个 EKS 集群在其他 VPC 上运行。在此示例中，VPC 对等关系可在 EKS 集群的 VPC 与连接器和 Cloud Volumes ONTAP 的 VPC 之间提供连接。



设置 RBAC 授权

您需要在每个 Kubernetes 集群上授权 Connector 角色，以便 Connector 可以发现和管理集群。

要启用不同的功能，需要不同的授权。

备份和还原

备份和还原只需要基本授权。

添加存储类

要使用 Cloud Manager 添加存储类，需要扩展授权。

安装 **Astra Trident**

要安装 Astra Trident，您需要为 Cloud Manager 提供完全授权。



安装 Astra Trident 时，Cloud Manager 会安装 Astra Trident 后端和 Kubernetes 密钥，其中包含 Astra Trident 与存储集群通信所需的凭据。

步骤

1. 创建集群角色和角色绑定。
 - a. 根据您的授权要求创建包含以下文本的 YAML 文件。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

```
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

```

存储类

添加扩展授权以使用 Cloud Manager 添加存储类。

```

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

```

安装 Trident

使用命令行提供完全授权并启用 Cloud Manager 以安装 Astra Trident 。

```

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

```

b. 将配置应用于集群。

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

2. 使用 `eksctl` 创建与权限组的标识映射。下面提供了一个示例。

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

"有关完整说明，请参见 [eksctl 文档](#)"。

Azure 中 Kubernetes 集群的要求

您可以使用 Cloud Manager 在 Azure 中添加和管理受管 Azure Kubernetes 集群（AKS）和自管 Kubernetes 集群。在将集群添加到 Cloud Manager 之前，请确保满足以下要求。

本主题使用 *Kubernetes cluster*，其中对于 AKS 和自管理 Kubernetes 集群的配置相同。在配置不同的位置指定集群类型。

要求

Astra Trident

需要使用四个最新版本的 Astra Trident 之一。您可以直接从 Cloud Manager 安装 Astra Trident。您应该 [查看前提条件](#) 安装 Astra Trident 之前。

要升级 Astra Trident，["使用操作员升级"](#)。

Cloud Volumes ONTAP

必须将 Cloud Volumes ONTAP 设置为集群的后端存储。["有关配置步骤，请转至 Astra Trident 文档"](#)。

Cloud Manager Connector

Connector 必须使用所需权限在 Azure 中运行。[在下方了解更多信息](#)。

网络连接

Kubernetes 集群和 Connector 之间以及 Kubernetes 集群和 Cloud Volumes ONTAP 之间需要网络连接。[在下方了解更多信息](#)。

RBAC 授权

Cloud Manager 支持使用和不使用 Active Directory 的已启用 RBAC 的集群。必须在每个 Azure 集群上授权 Cloud Manager Connector 角色。[在下方了解更多信息](#)。

准备连接器

要发现和管理 Kubernetes 集群，需要使用 Azure 中的 Cloud Manager Connector。您需要创建新的 Connector 或使用具有所需权限的现有 Connector。

创建新的 Connector

按照以下链接之一中的步骤进行操作。

- ["从 Cloud Manager 创建 Connector"](#) 建议
- ["从 Azure Marketplace 创建 Connector"](#)
- ["在现有 Linux 主机上安装 Connector"](#)

向现有 **Connector** 添加所需权限（以发现受管 **AKS** 集群）

如果要发现受管 AKS 集群，您可能需要修改 Connector 的自定义角色以提供权限。

步骤

1. 确定分配给 Connector 虚拟机的角色：
 - a. 在 Azure 门户中，打开虚拟机服务。
 - b. 选择 Connector 虚拟机。
 - c. 在设置下，选择 * 身份 *。
 - d. 单击 * Azure 角色分配 *。
 - e. 记下分配给 Connector 虚拟机的自定义角色。
2. 更新自定义角色：
 - a. 在 Azure 门户中，打开 Azure 订阅。
 - b. 单击 * 访问控制（IAM） > 角色 *。
 - c. 单击自定义角色的省略号（...），然后单击 * 编辑 *。
 - d. 单击 JSON 并添加以下权限：

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

- e. 单击 * 查看 + 更新 *，然后单击 * 更新 *。

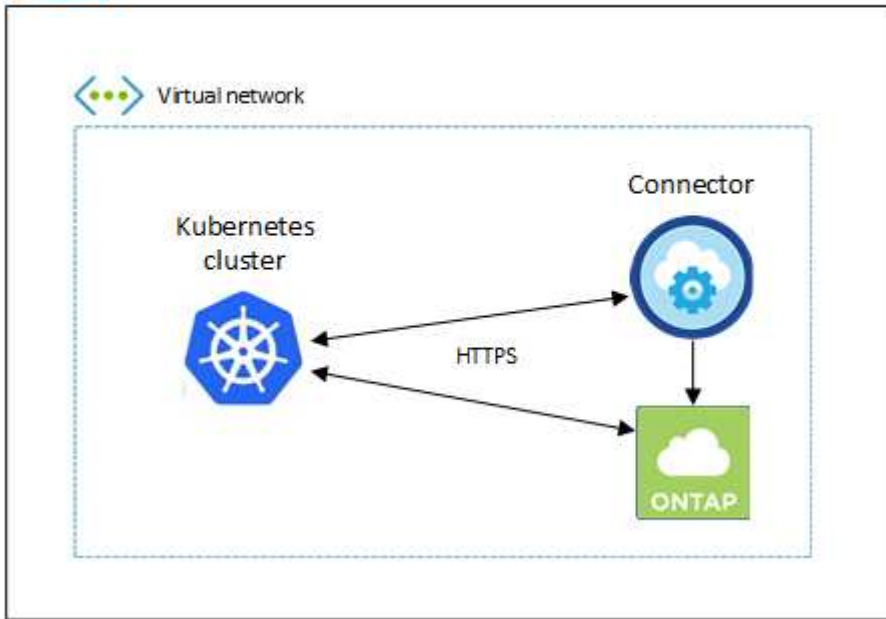
查看网络连接要求

您需要在 Kubernetes 集群和 Connector 之间以及 Kubernetes 集群与为集群提供后端存储的 Cloud Volumes ONTAP 系统之间提供网络连接。

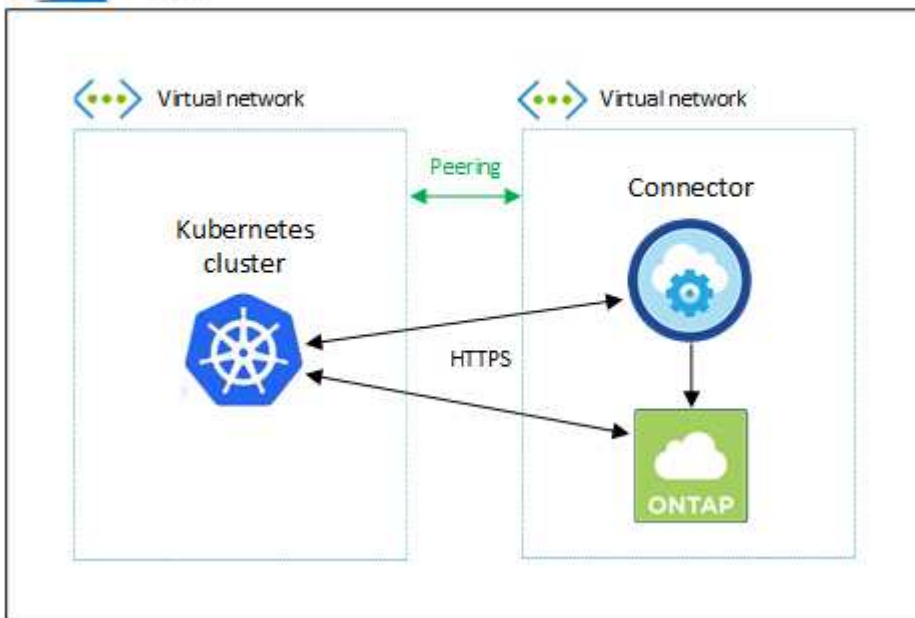
- 每个 Kubernetes 集群都必须与 Connector 建立入站连接
- 此连接器必须通过端口 443 与每个 Kubernetes 集群建立出站连接

提供此连接的最简单方法是，将 Connector 和 Cloud Volumes ONTAP 部署在与 Kubernetes 集群相同的 VNet 中。否则，您需要在不同的 VN 集之间设置对等连接。

以下示例显示了同一 vNet 中的每个组件。



下面是另一个示例，其中显示了一个 Kubernetes 集群运行在另一个 vNet 中。在此示例中，对等关系可在 Kubernetes 集群的 vNet 与 Connector 和 Cloud Volumes ONTAP 的 vNet 之间建立连接。



设置 RBAC 授权

RBAC 验证仅在启用了 Active Directory (AD) 的 Kubernetes 集群上进行。不带 AD 的 Kubernetes 集群将自动通过验证。

您需要在每个 Kubernetes 集群上授权 Connector 角色，以便 Connector 可以发现和管理集群。

备份和还原

备份和还原只需要基本授权。

添加存储类

要使用 Cloud Manager 添加存储类，需要扩展授权。

安装 Astra Trident

要安装 Astra Trident，您需要为 Cloud Manager 提供完全授权。



安装 Astra Trident 时，Cloud Manager 会安装 Astra Trident 后端和 Kubernetes 密钥，其中包含 Astra Trident 与存储集群通信所需的凭据。

您的 RBAC 对象： name : 配置会根据您的 Kubernetes 集群类型稍有不同。

- 如果要部署 * 受管 AKS 集群 *，则需要为 Connector 的系统分配的受管身份提供对象 ID。此 ID 可在 Azure 管理门户中使用。

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. The ID field is highlighted with a red rectangle and contains the GUID '0c288856-adea-485b-a4dc-c15b5ce2c401'. Below this is a 'Permissions' section with a button for 'Azure role assignments'.

- 如果要部署 * 自管理 Kubernetes 集群 *，则需要任何授权用户的用户名。

创建集群角色和角色绑定。

1. 根据您的授权要求创建包含以下文本的 YAML 文件。将 `subjects : kind :` 变量替换为您的用户名，将 `subjects : user :` 替换为系统分配的受管身份的对象 ID 或上述任何授权用户的用户名。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

```
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

```

存储类

添加扩展授权以使用 Cloud Manager 添加存储类。

```

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

```

安装 Trident

使用命令行提供完全授权并启用 Cloud Manager 以安装 Astra Trident 。

```

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

```

2. 将配置应用于集群。

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

Google Cloud 中的 Kubernetes 集群的要求

您可以使用 Cloud Manager 在 Google 中添加和管理受管 Google Kubernetes Engine (GKEE) 集群和自管 Kubernetes 集群。在将集群添加到 Cloud Manager 之前，请确保满足以下要求。

本主题使用 *Kubernetes cluster*，其中 GKE- 和自管理 Kubernetes 集群的配置相同。在配置不同的位置指定集群类型。

要求

Astra Trident

需要使用四个最新版本的 Astra Trident 之一。您可以直接从 Cloud Manager 安装 Astra Trident。您应该 ["查看前提条件"](#) 安装 Astra Trident 之前

要升级 Astra Trident，["使用操作员升级"](#)。

Cloud Volumes ONTAP

Cloud Volumes ONTAP 必须与 Kubernetes 集群位于同一租户帐户，工作空间和连接器下的 Cloud Manager 中。["有关配置步骤，请转至 Astra Trident 文档"](#)。

Cloud Manager Connector

必须使用所需权限在 Google 中运行 Connector。[在下方了解更多信息。](#)

网络连接

Kubernetes 集群和 Connector 之间以及 Kubernetes 集群和 Cloud Volumes ONTAP 之间需要网络连接。[在下方了解更多信息。](#)

RBAC 授权

Cloud Manager 支持使用和不使用 Active Directory 的已启用 RBAC 的集群。必须在每个 GKE 集群上授权 Cloud Manager Connector 角色。[在下方了解更多信息。](#)

准备连接器

要发现和管理 Kubernetes 集群，需要使用 Google 中的 Cloud Manager Connector。您需要创建新的 Connector 或使用具有所需权限的现有 Connector。

创建新的 Connector

按照以下链接之一中的步骤进行操作。

- ["从 Cloud Manager 创建 Connector"](#) 建议
- ["在现有 Linux 主机上安装 Connector"](#)

将所需权限添加到现有 **Connector**（以发现受管 **GKEE** 集群）

如果要发现受管 GKEE 集群，您可能需要修改 Connector 的自定义角色以提供权限。

步骤

1. 在中 ["云控制台"](#)下，转到 * 角色 * 页面。
2. 使用页面顶部的下拉列表，选择包含要编辑的角色的项目或组织。
3. 单击一个自定义角色。
4. 单击 * 编辑角色 * 以更新角色的权限。
5. 单击 * 添加权限 * 向角色添加以下新权限。

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

6. 单击 * 更新 * 以保存已编辑的角色。

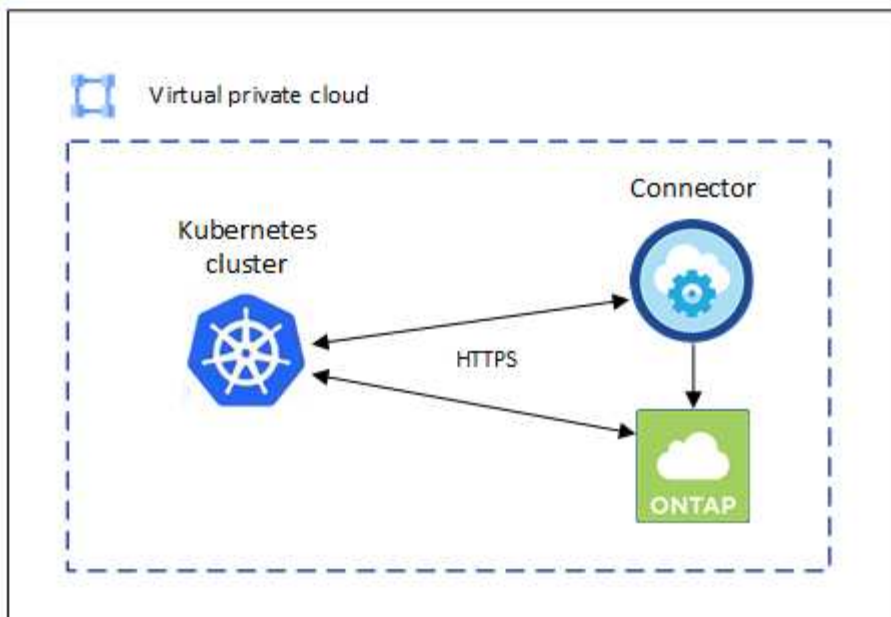
查看网络连接要求

您需要在 Kubernetes 集群和 Connector 之间以及 Kubernetes 集群与为集群提供后端存储的 Cloud Volumes ONTAP 系统之间提供网络连接。

- 每个 Kubernetes 集群都必须与 Connector 建立入站连接
- 此连接器必须通过端口 443 与每个 Kubernetes 集群建立出站连接

提供此连接的最简单方法是，将连接器和 Cloud Volumes ONTAP 部署在与 Kubernetes 集群相同的 VPC 中。否则，您需要在不同的 VPC 之间设置对等连接。

以下示例显示了同一 VPC 中的每个组件。



设置 RBAC 授权

RBAC 验证仅在启用了 Active Directory (AD) 的 Kubernetes 集群上进行。不带 AD 的 Kubernetes 集群将自动通过验证。

您需要在每个 Kubernetes 集群上授权 Connector 角色，以便 Connector 可以发现和管理集群。

备份和还原

备份和还原只需要基本授权。

添加存储类

要使用 Cloud Manager 添加存储类，需要扩展授权。

安装 Astra Trident

要安装 Astra Trident，您需要为 Cloud Manager 提供完全授权。



安装 Astra Trident 时，Cloud Manager 会安装 Astra Trident 后端和 Kubernetes 密钥，其中包含 Astra Trident 与存储集群通信所需的凭据。

要在 YAML 文件中配置子对象： `name :`，您需要知道 Cloud Manager 的唯一 ID。

您可以通过以下两种方式之一找到唯一 ID：

- 使用命令：

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

- 在上的服务帐户详细信息中 "云控制台"。

The screenshot shows the 'Cloud Manager Service Account' page in the 'CloudSync-Dev' console. The page has a blue header with the console name and a back arrow. Below the header is a navigation bar with tabs: DETAILS, PERMISSIONS, KEYS, METRICS, and LOGS. The 'DETAILS' tab is selected. The main content area is titled 'Service account details' and contains three sections: 'Name' with a text input field containing 'Cloud Manager Service Account' and a 'SAVE' button; 'Description' with a text input field and a 'SAVE' button; and 'Email' with a text input field containing 'cloudmanager-service-account@cloudsync-dev-214020.iam.gserviceaccount.com'. Below the email field is the 'Unique ID' section, which displays the ID '102217358851946603445' in a yellow highlighted box.

创建集群角色和角色绑定。

1. 根据您的授权要求创建包含以下文本的 YAML 文件。将 `ssubforjects : kind :` 变量替换为您的用户名, 将 `ssubforjects : user :` 替换为授权服务帐户的唯一 ID 。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

```
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

```

存储类

添加扩展授权以使用 Cloud Manager 添加存储类。

```

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

```

安装 Trident

使用命令行提供完全授权并启用 Cloud Manager 以安装 Astra Trident。

```

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

```

2. 将配置应用于集群。


```
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.