要求

Kubernetes clusters

NetApp
April 16, 2024

目录

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BlueXP。在将集群添加到BlueXP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BlueXP安装或升级Astra Trident。您应该 "查看前提条件" 安装 Astra Trident 之前。

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"。

BlueXP连接器

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BlueXP 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BlueXP 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。

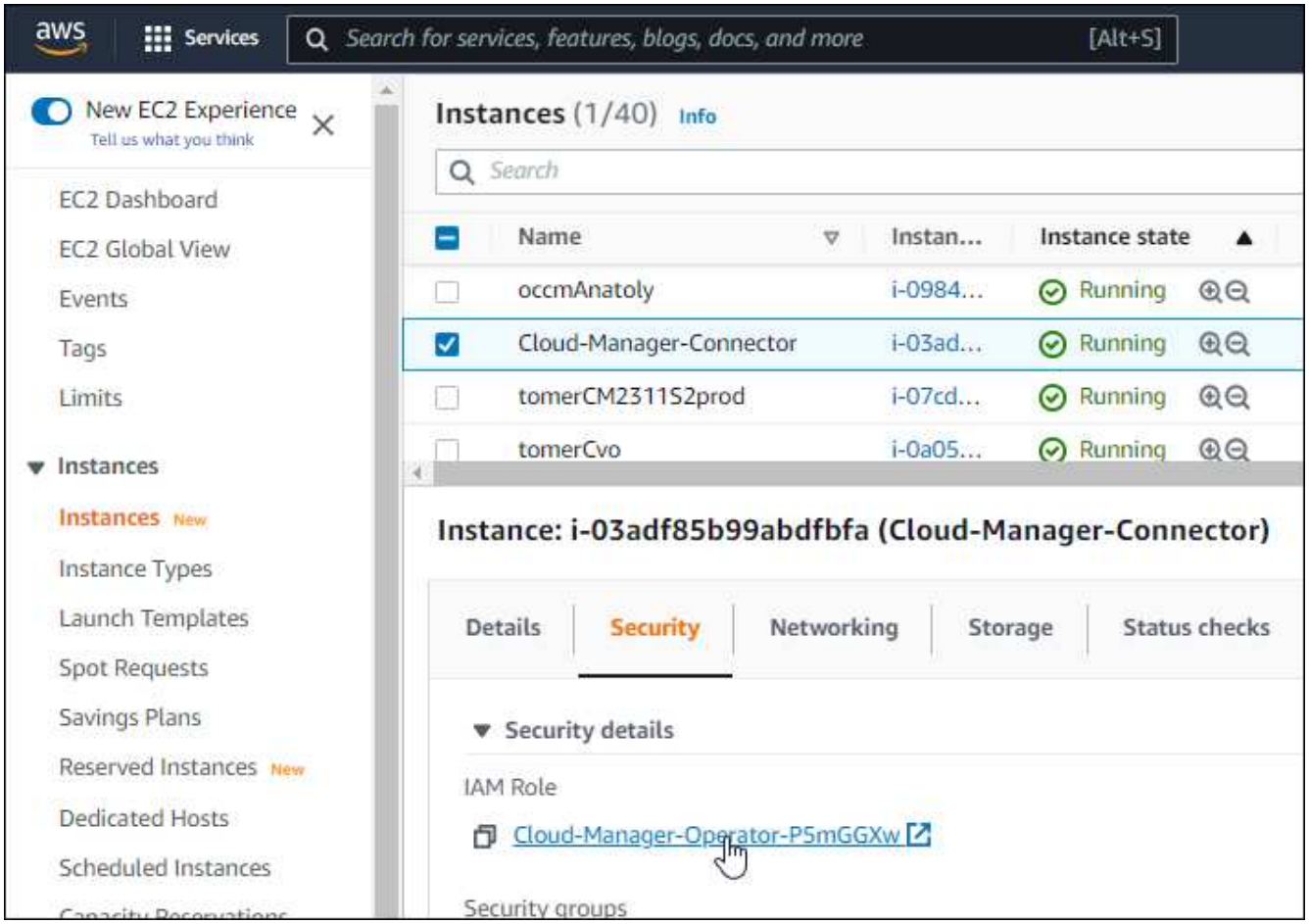
- "从BlueXP创建连接器" 建议
- "从 AWS Marketplace 创建 Connector"
- "在 AWS 的现有 Linux 主机上安装 Connector"

将所需权限添加到现有 **Connector**

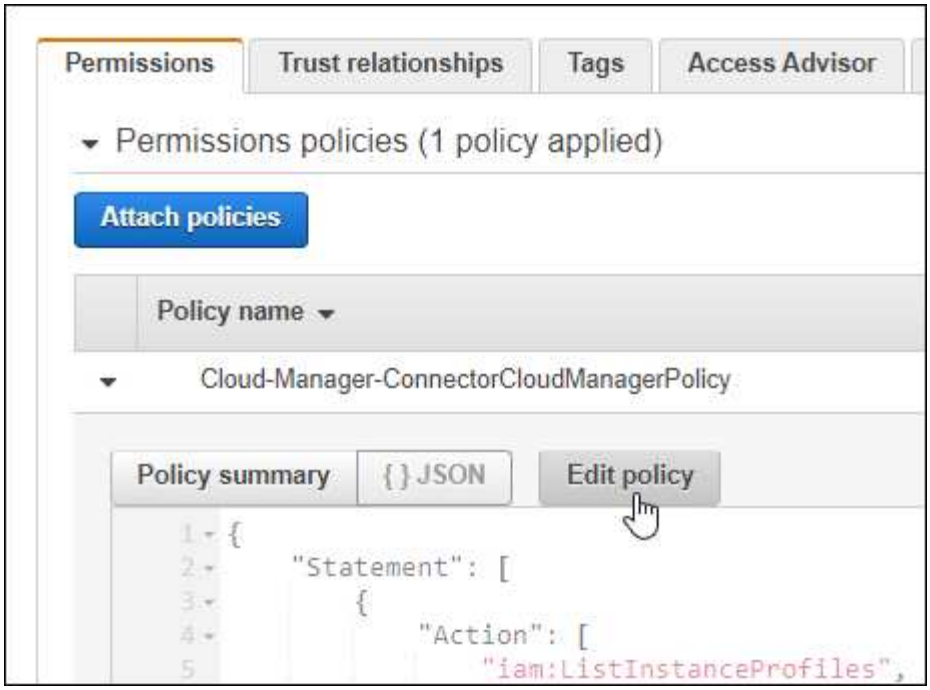
从 3.9.13 版开始，任何 *new* 创建的 Connectors 均包含三个新的 AWS 权限，用于发现和管理 Kubernetes 集群。如果您在此版本之前创建了 Connector ，则需要修改此 Connector 的 IAM 角色的现有策略以提供权限。

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：

   - EC2：Describe注册
   - EKS：ListClusters
   - EKS：Describe集群
   - 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 * 。

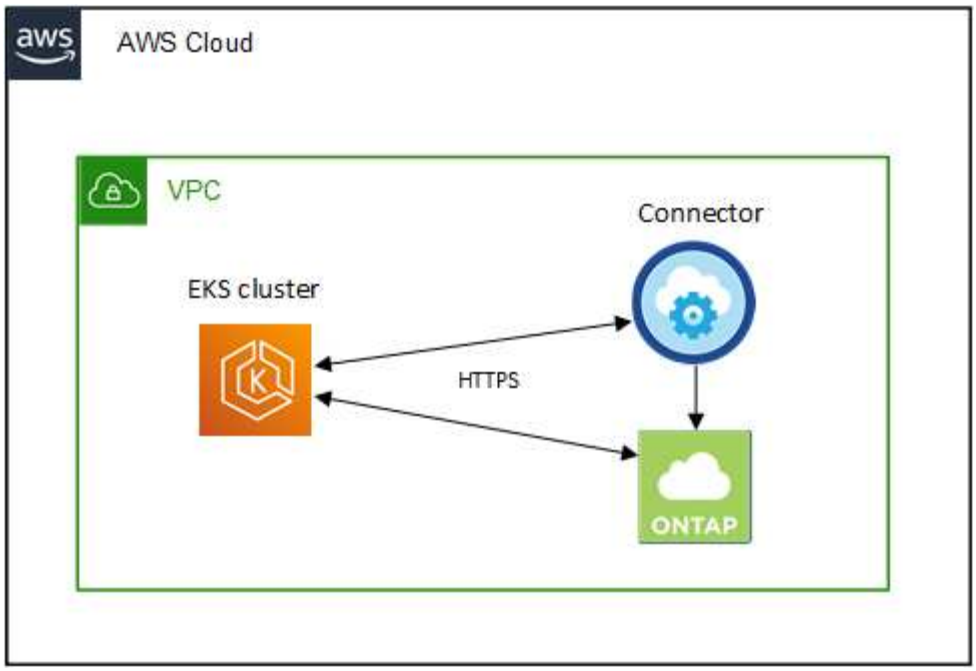
查看网络连接要求

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。

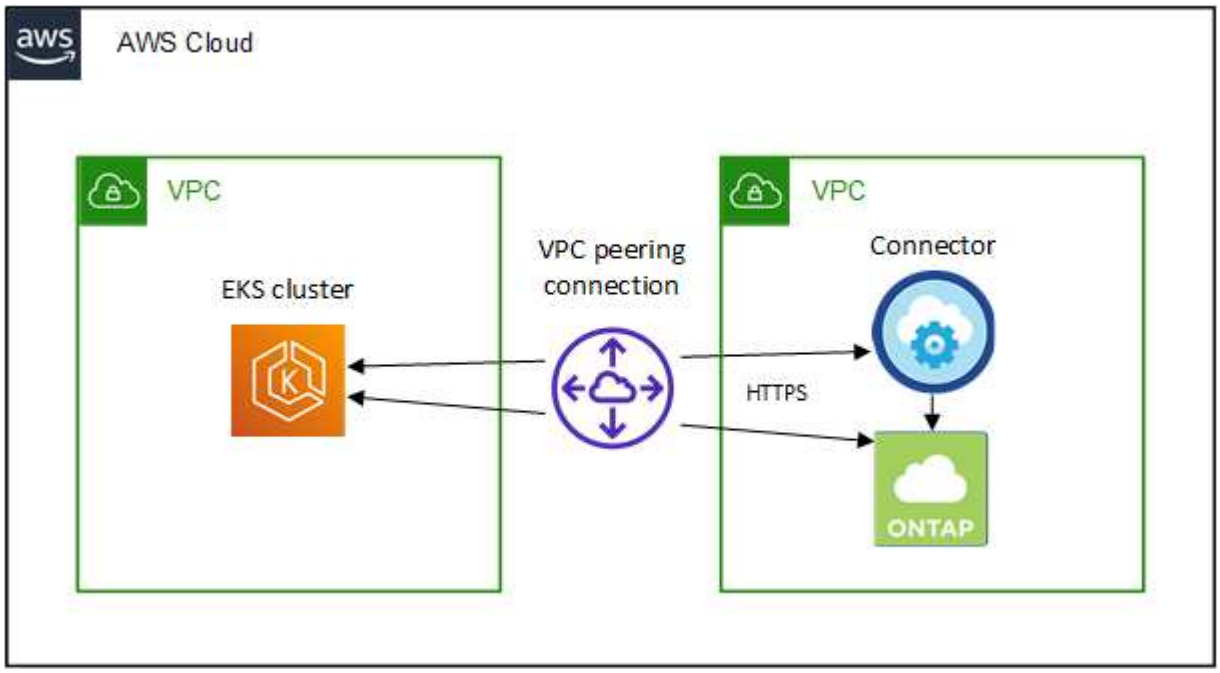
- 每个 Kubernetes 集群都必须与 Connector 建立入站连接
- 此连接器必须通过端口 443 与每个 Kubernetes 集群建立出站连接

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BlueXP添加存储类并监控集群中的后端更改、需要扩展授权。

安装 **Astra Trident**

要安装Astra Trident、您需要为BlueXP提供完全授权。

> 安装Astra Trident时、BlueXP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. 您可以根据需要自定义授权。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

```yaml
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
```

存储类

添加扩展授权以使用BlueXP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
          - 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
```

安装 {\f270通过} {\f151。}

使用命令行提供完全授权并启用BlueXP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. 创建与权限组的标识映射。

使用 **eksctl**

使用eksctl在集群与BlueXP Connector的IAM角色之间创建IAM身份映射。

"有关完整说明，请参见 eksctl 文档"。

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}}
```

编辑 **AWS-auth**

直接编辑AWS-auth ConfigMap、以便为BlueXP Connector的IAM角色添加RBAC访问权限。

"有关完整说明，请参见 AWS EKS 文档"。

下面提供了一个示例。

```
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 中 Kubernetes 集群的要求

您可以使用BlueXP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BlueXP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BlueXP安装或升级Astra Trident。您应该 "查看前提条件" 安装 Astra Trident 之前。

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"。

BlueXP连接器

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

无论是否使用Active Directory、BlueXP都支持启用了RBAC的集群。必须在每个Azure集群上授权BlueXP 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BlueXP 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BlueXP创建连接器" 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 * 。

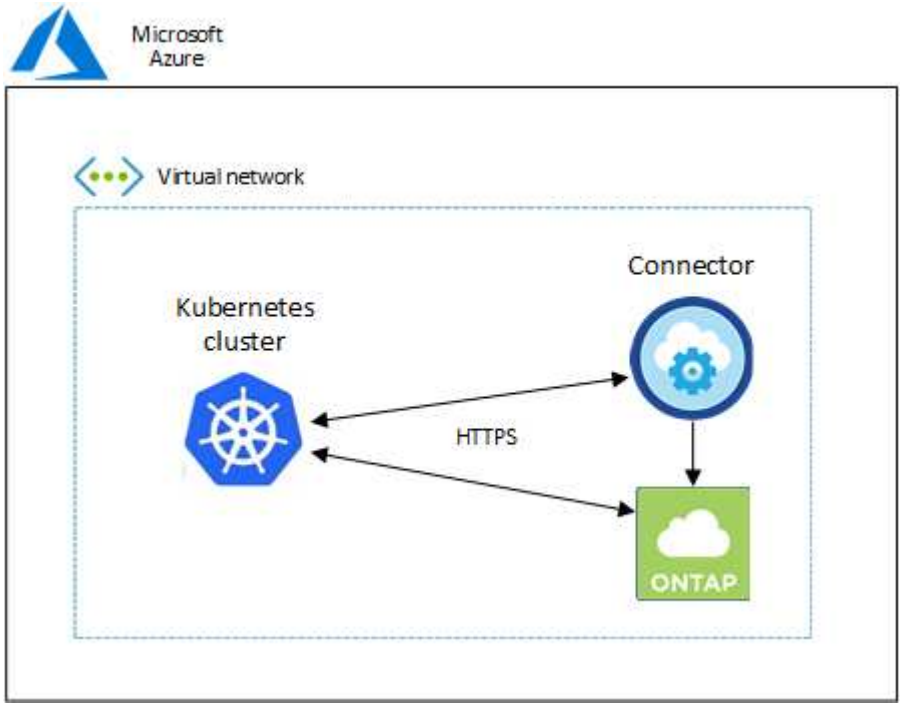
查看网络连接要求

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。

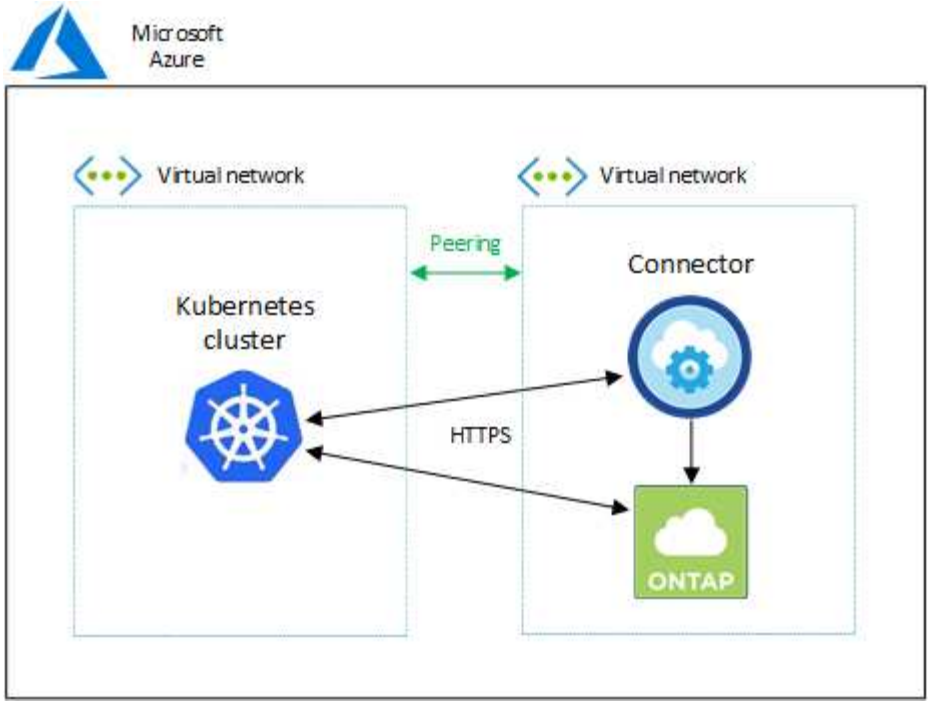
- 每个 Kubernetes 集群都必须与 Connector 建立入站连接
- 此连接器必须通过端口 443 与每个 Kubernetes 集群建立出站连接

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BlueXP添加存储类并监控集群中的后端更改、需要扩展授权。

安装 **Astra Trident**

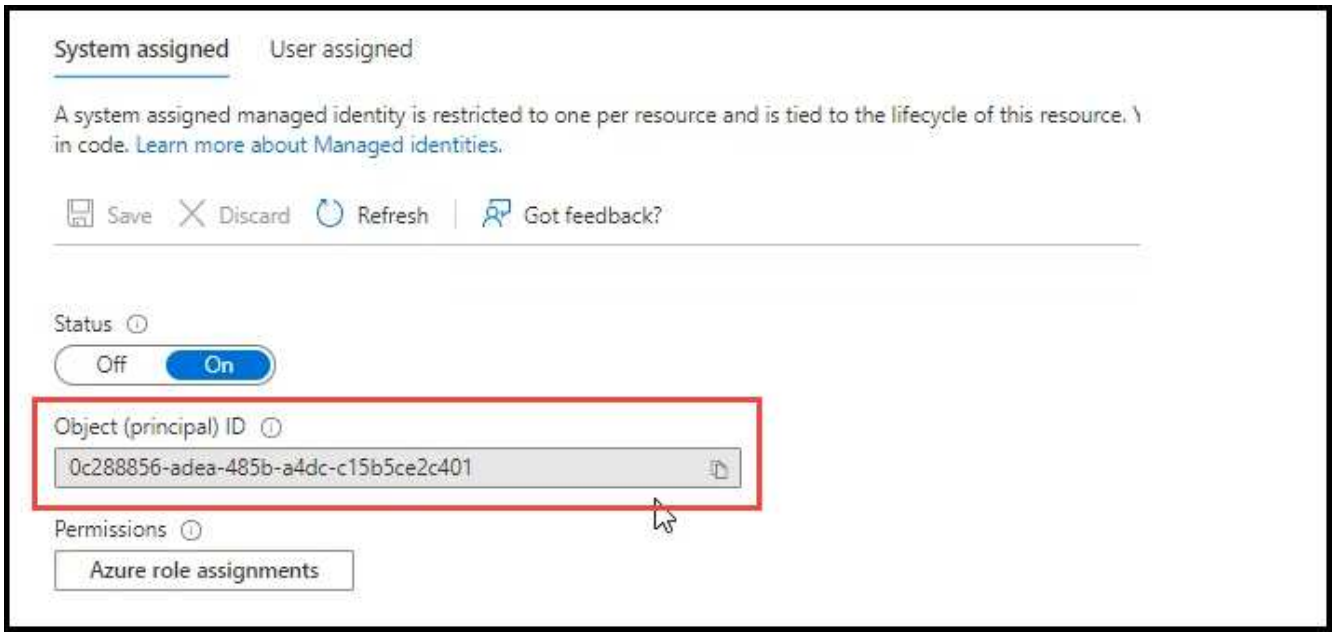
要安装Astra Trident、您需要为BlueXP提供完全授权。

> 安装Astra Trident时、BlueXP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。

开始之前

您的 RBAC `s`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。



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

步骤

创建集群角色和角色绑定。

1. 您可以根据需要自定义授权。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

更换 `subjects: kind:` 变量、并输入您的用户名和 `subjects: name:` 具有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。

```yaml
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
```

存储类

添加扩展授权以使用BlueXP添加存储类。

更换 `subjects: kind:` 变量、并输入您的用户名和 `subjects: user:` 具有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。

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

```yaml
            - 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
```

安装 {\f270通过} {\f151。}

使用命令行提供完全授权并启用BlueXP以安装Astra Trident。

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

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BlueXP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BlueXP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BlueXP安装或升级Astra Trident。您应该 "查看前提条件" 安装 Astra Trident 之前

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BlueXP中。 "有关配置步骤，请转至 Astra Trident 文档"。

BlueXP连接器

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

无论是否使用Active Directory、BlueXP都支持启用了RBAC的集群。必须在每个GKE集群上授权BlueXP 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BlueXP 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BlueXP创建连接器" 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。

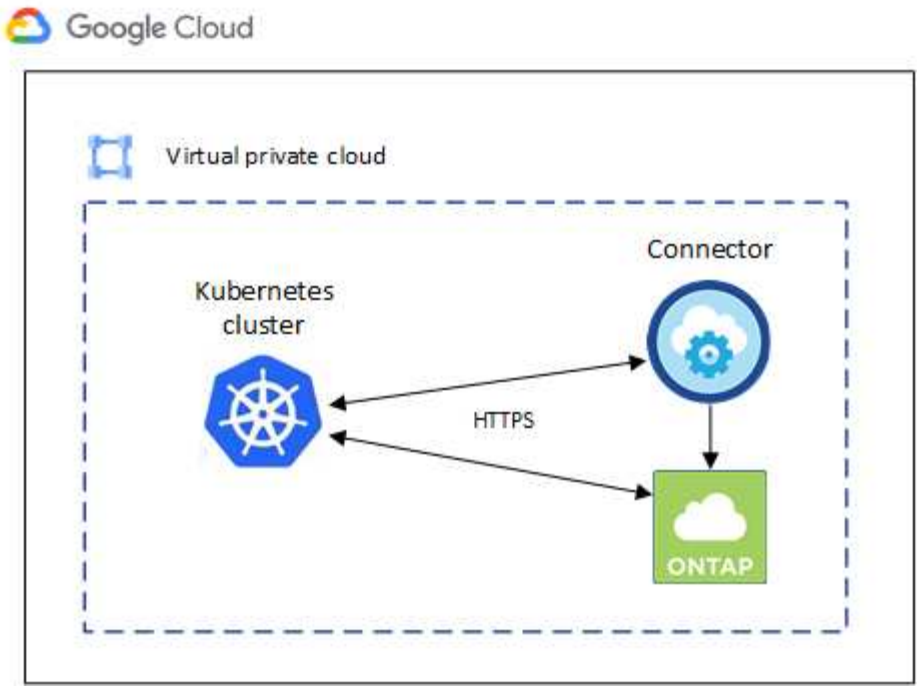
查看网络连接要求

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BlueXP添加存储类并监控集群中的后端更改、需要扩展授权。

安装 **Astra Trident**

要安装Astra Trident、您需要为BlueXP提供完全授权。

> 安装Astra Trident时、BlueXP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。

开始之前

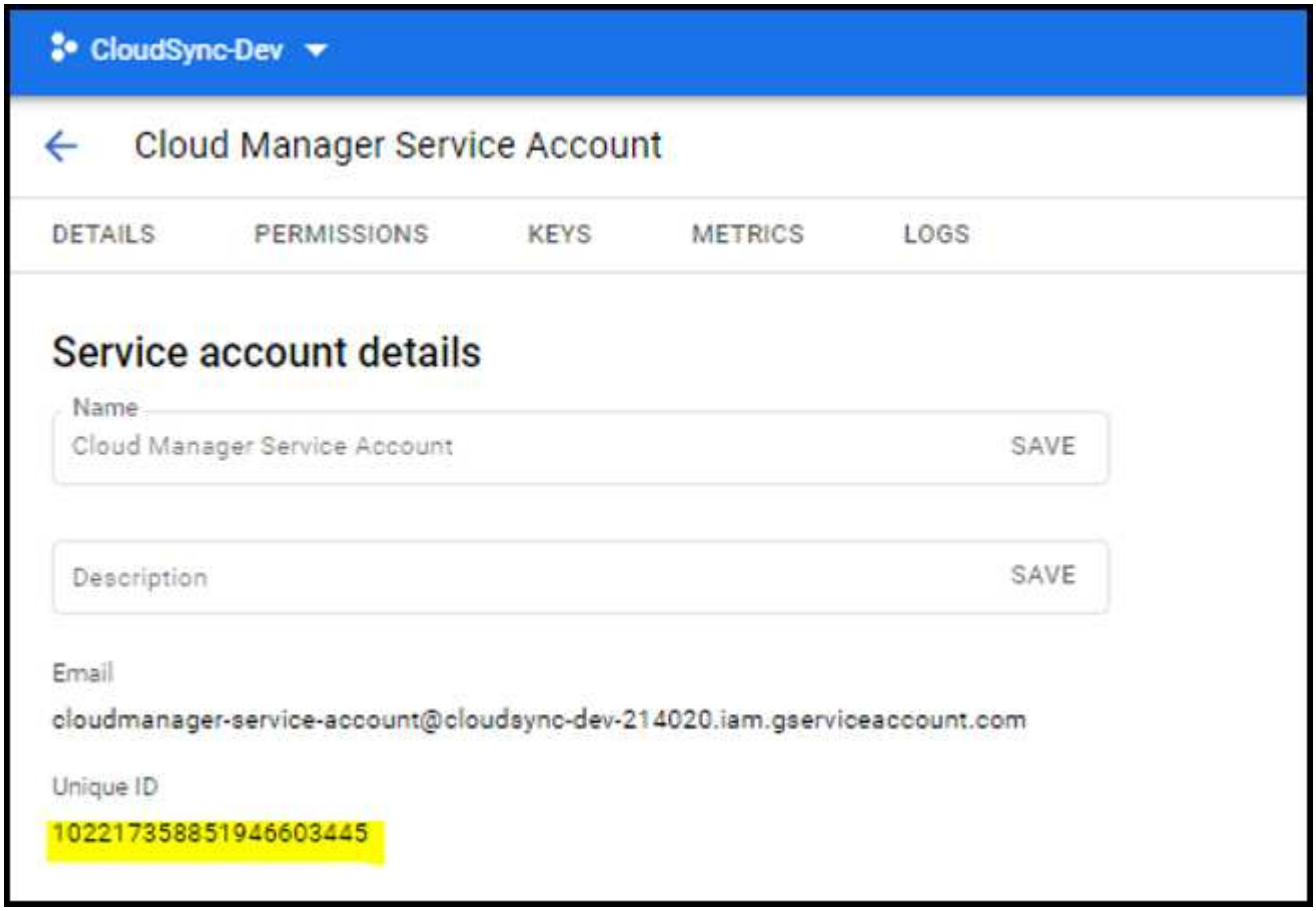
以进行配置 `subjects: name:` 在YAML文件中、您需要知道BlueXP的唯一ID。

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"。



步骤

创建集群角色和角色绑定。

1. 您可以根据需要自定义授权。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

更换 `subjects: kind:` 变量、并输入您的用户名和 `subjects: name:` 具有授权服务帐户的唯一ID。

```
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
```

存储类

添加扩展授权以使用BlueXP添加存储类。

更换 `subjects: kind:` 变量、并输入您的用户名和 `subjects: user:` 具有授权服务帐户的唯一ID。

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

```yaml
            - 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
```

安装 {**\f270**通过} {**\f151**。}

使用命令行提供完全授权并启用BlueXP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>
```

OpenShift中的Kubernetes集群的要求

您可以使用BlueXP添加和管理自管理OpenShift Kubernetes集群。在将集群添加到BlueXP之前、请确保满足以下要求。

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BlueXP安装或升级Astra Trident。您应该 "查看前提条件" 安装 Astra Trident 之前。

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"。

BlueXP连接器

要导入和管理Kubernetes集群、需要使用BlueXP Connector。您需要创建新的Connector或使用具有云提供商所需权限的现有Connector：

- "AWS连接器"
- "Azure连接器"
- "Google Cloud Connector"

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。

具有**RBAC**授权的**Kubernetes**配置文件**(kubeconfig)**

要导入OpenShift集群、您需要一个kubeconfig文件、该文件具有启用不同功能所需的RBAC授权。 [创

建kubeconfig文件]。

- 备份和还原：备份和还原仅需要基本授权。
- 添加存储类：要使用BlueXP添加存储类并监控集群中对后端的更改、需要扩展授权。
- 安装Astra Trident：要安装Astra Trident、您需要为BlueXP提供完全授权。

> 安装Astra Trident时、BlueXP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。

创建kubeconfig文件

使用OpenShift命令行界面创建一个kubeconfig文件以导入到BlueXP中。

步骤

1. 使用管理员用户通过公有 URL使用`oc login`登录到OpenShift命令行界面。
2. 按如下所示创建服务帐户：

   a. 创建名为`oc-service-account.yaml`的服务帐户文件。

   根据需要调整名称和命名空间。如果在此处进行了更改，则应在以下步骤中应用相同的更改。

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

+

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

   a. 应用服务帐户：

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

3. 根据您的授权要求创建自定义角色绑定。

   a. 创建名为`oc-clusterrolebind.yaml`的`ClusterRoleBindingm`文件。

```
oc-clusterrolebinding.yaml
```

   b. 根据需要为集群配置RBAC授权。

备份 / 还原

添加基本授权，以便为 Kubernetes 集群启用备份和还原。

```yaml
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
```

存储类

添加扩展授权以使用BlueXP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
          - 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
```

安装 {\f270通过} {\f151。}

授予完全管理员授权并启用BlueXP以安装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. 应用集群角色绑定：

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

4. 列出服务帐户密码，将 ` <context>` 替换为适用于您的安装的正确上下文：

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

输出的结尾应类似于以下内容：

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

`sec`白 烟 数组中每个元素的索引均以 0 开头。在上面的示例中、`oc-service-account-dockercfg-vhz87`的索引为0、`oc-service-account-token-r59krs`的索引为1。在输出中，记下包含 "token" 一词的服务帐户名称的索引。

5. 按如下所示生成 kubeconfig ：

a. 创建 `create-kubeconfig.sh` 文件。将以下脚本开头的 `token_index` 替换为正确的值。

```
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. 获取用于将其应用于 Kubernetes 集群的命令。

```
source create-kubeconfig.sh
```

结果

您将使用生成的 `kubeconfig-sa` 用于将OpenShift集群添加到BlueXP的文件。