需求
Kubernetes clusters

NetApp
April 16, 2024

目錄

需求

AWS中Kubernetes叢集的需求

您可以將AWS上的託管Amazon Elastic Kubernetes Service（EKS）叢集或自我管理Kubernetes叢集新增至BlueXP。在將叢集新增至BlueXP之前、您必須確保符合下列需求。

> 本主題使用_Kubernetes叢集_、其中EKS和自我管理Kubernetes叢集的組態相同。叢集類型是在組態不同的地方指定。

需求

Astra Trident

需要最新版Astra Trident的四種版本之一。您可以直接從BlueXP安裝或升級Astra Trident。您應該 "檢閱先決條件" 安裝Astra Trident之前。

Cloud Volumes ONTAP

AWS的for AWS必須設定為叢集的後端儲存設備。Cloud Volumes ONTAP "如需組態步驟、請前往Astra Trident文件"。

BlueXP Connector

連接器必須以所需權限在AWS中執行。 深入瞭解。

網路連線能力

Kubernetes叢集和Connector之間、以及Kubernetes叢集和Cloud Volumes ONTAP 整個過程之間、都需要網路連線。 深入瞭解。

RBAC授權

每個Kubernetes叢集都必須授權BlueXP Connector角色。 深入瞭解。

準備連接器

AWS需要使用BlueXP Connector來探索及管理Kubernetes叢集。您需要建立新的Connector、或是使用具有所需權限的現有Connector。

建立新的**Connector**

請遵循下列其中一個連結中的步驟。

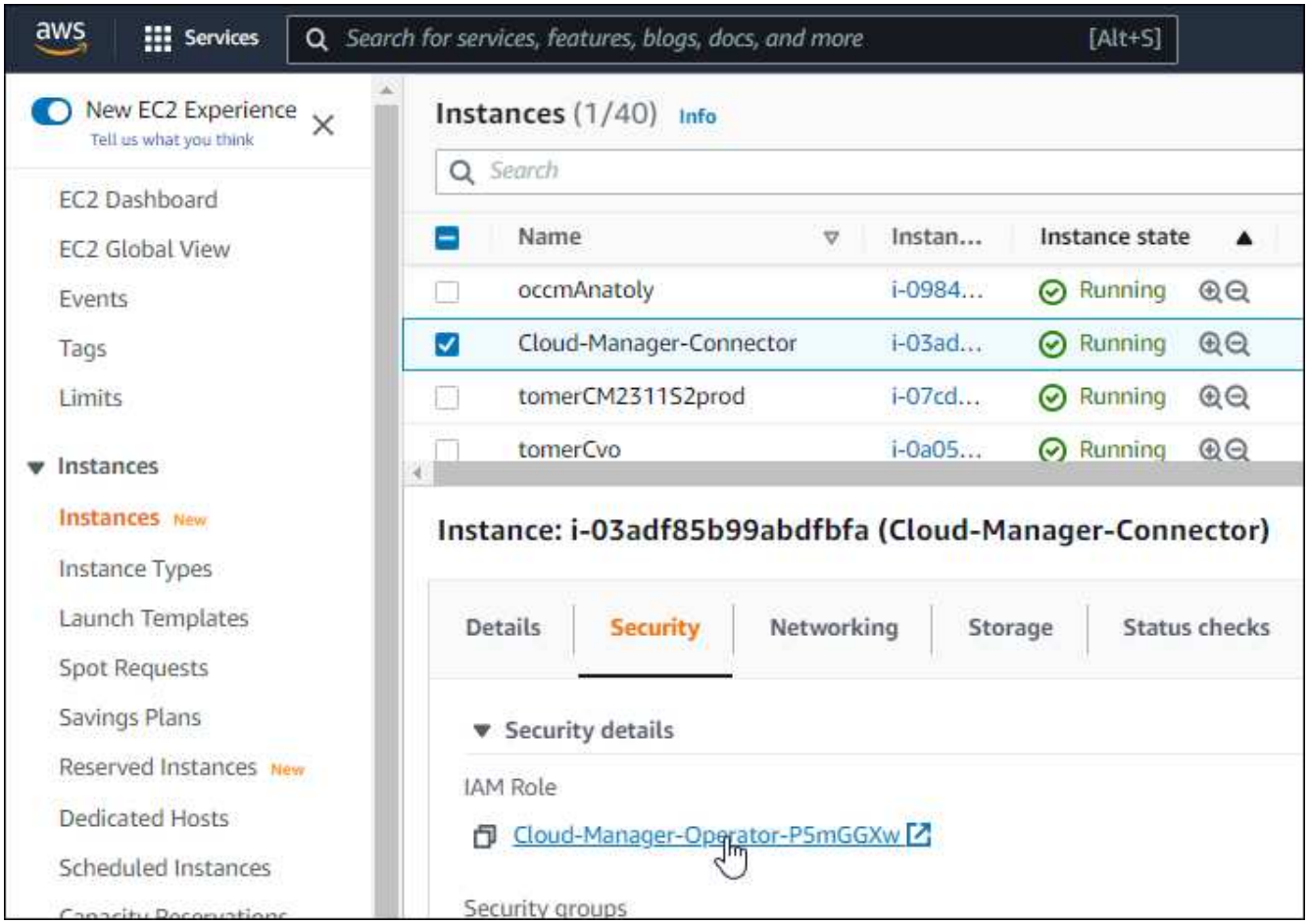
- "從BlueXP建立連接器" （建議）
- "從AWS Marketplace建立連接器"
- "在AWS中現有的Linux主機上安裝Connector"

將必要的權限新增至現有的**Connector**

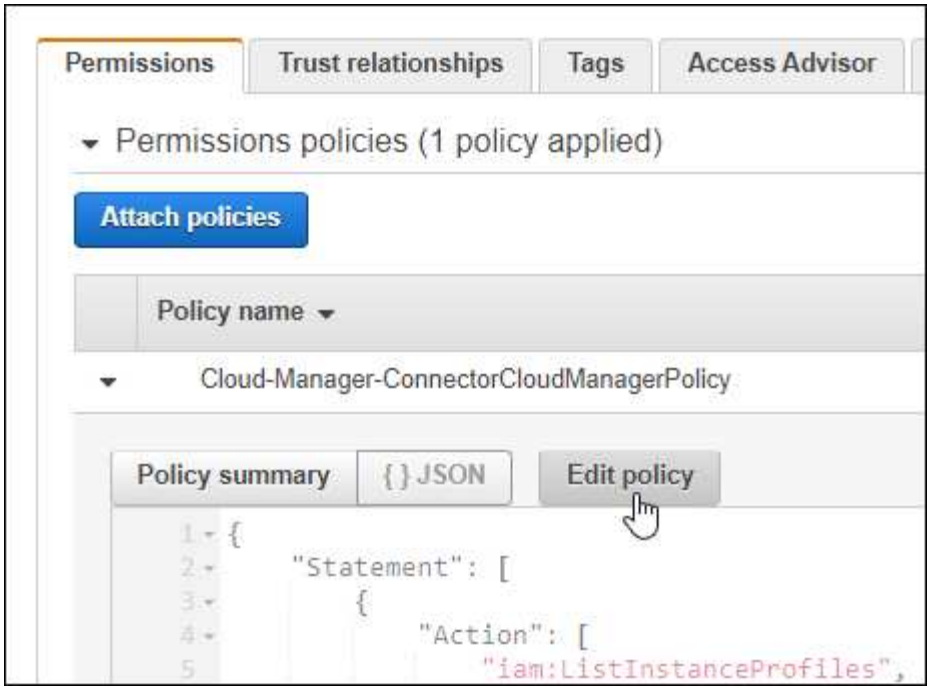
從3.9.13版開始、任何_new建立的連接器都包含三個新的AWS權限、可用來探索及管理Kubernetes叢集。如果您在此版本之前建立了Connector、則需要修改Connector IAM角色的現有原則、以提供權限。

步驟

1. 移至AWS主控台並開啟EC2服務。
2. 選取連接器執行個體、按一下*安全性*、然後按一下IAM角色名稱、即可檢視IAM服務中的角色。



3. 在「權限」索引標籤中、展開原則、然後按一下「編輯原則」。



4. 按一下「* JSON*」、然後在第一組動作下新增下列權限：
   - EC2：取消註冊
   - EKS：清單叢集
   - EKS：取消叢集
   - IAM：GetInstanceProfile

   "檢視原則的完整Json格式"

5. 按一下「檢閱原則」、然後按一下「儲存變更」。

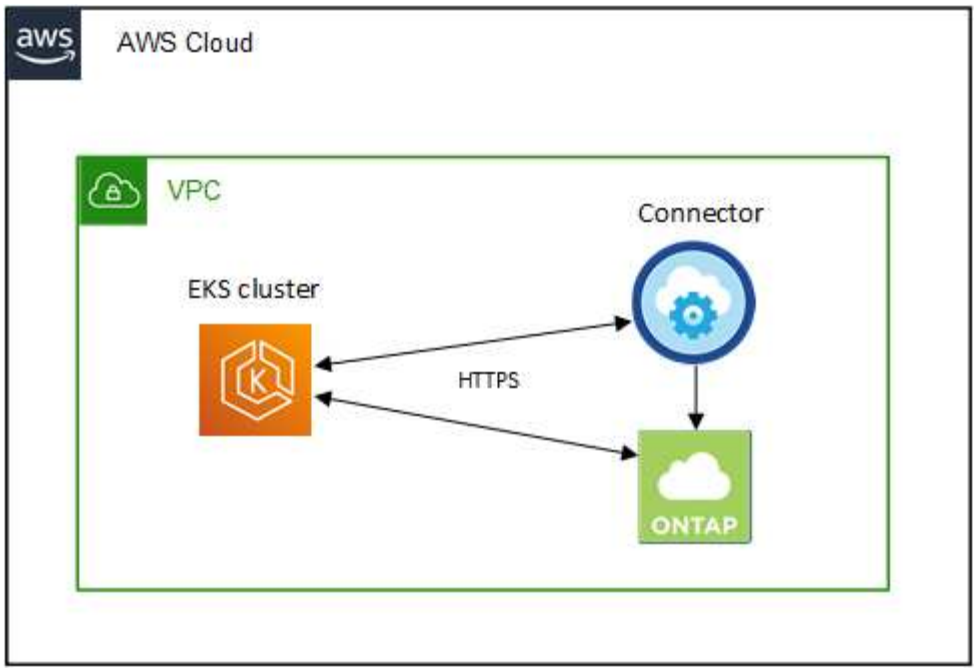
檢閱網路需求

您需要在Kubernetes叢集與Connector之間、以及Kubernetes叢集與Cloud Volumes ONTAP 為叢集提供後端儲存功能的支援系統之間、提供網路連線。

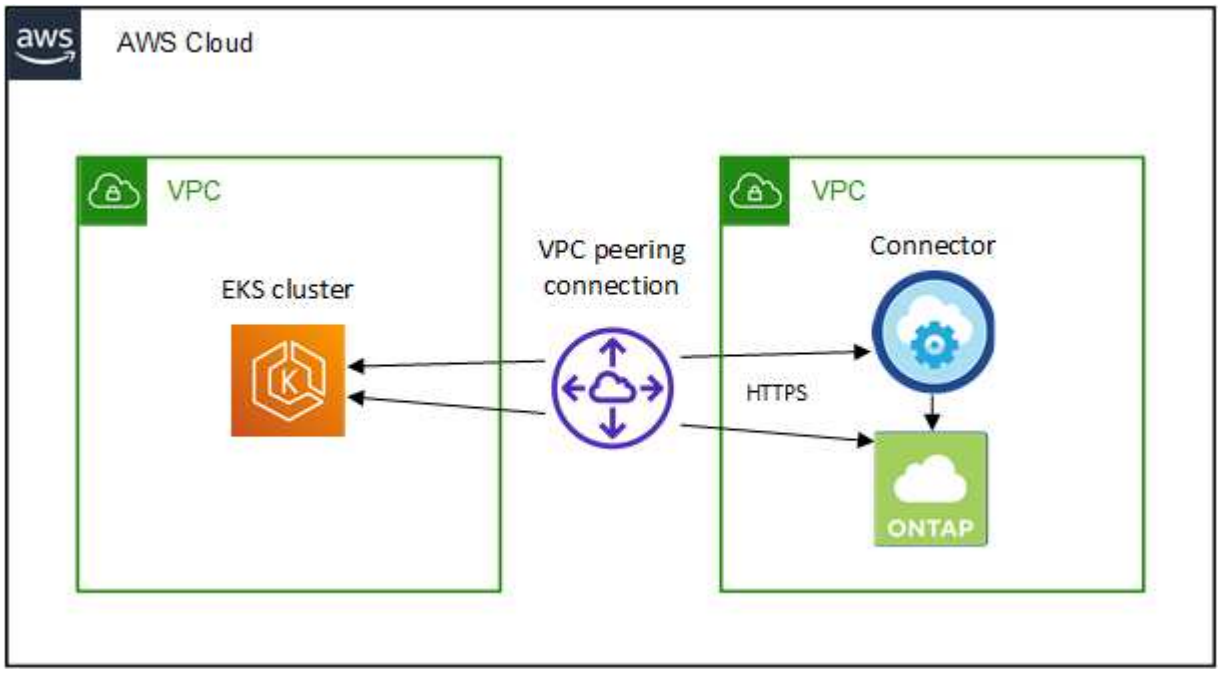
- 每個Kubernetes叢集都必須有來自Connector的傳入連線
- 連接器必須透過連接埠443連線至每個Kubernetes叢集

提供這種連線能力的最簡單方法、就是將Connector和Cloud Volumes ONTAP Sfor部署在Kubernetes叢集所在的VPC上。否則、您需要在不同的VPC之間設定VPC對等連線。

以下範例顯示同一VPC中的每個元件。



以下是另一個範例、顯示在不同VPC上執行的EKS叢集。在此範例中、VPC對等功能可在EKS叢集的VPC與連接器和Cloud Volumes ONTAP 物件的VPC之間建立連線。



設定RBAC授權

您需要在每個Kubernetes叢集上授權Connector角色、以便Connector能夠探索及管理叢集。

需要不同的授權才能啟用不同的功能。

備份與還原

備份與還原僅需基本授權。

新增儲存類別

若要使用BlueXP新增儲存類別、並監控叢集是否有變更後端、則需要擴充授權。

安裝**Astra Trident**

您必須提供BlueXP的完整授權、才能安裝Astra Trident。

> 安裝Astra Trident時、BlueXP會安裝Astra Trident後端和Kubernetes機密、其中包含Astra Trident與儲存叢集通訊所需的認證資料。

步驟

1. 建立叢集角色和角色繫結。
    a. 您可以根據自己的需求自訂授權。

備份/還原

新增基本授權以啟用Kubernetes叢集的備份與還原。

```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
```

儲存類別

新增擴充授權、以使用BlueXP新增儲存類別。

```
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
```

Trident 安装

使用命令列提供完整授權、並讓BlueXP安裝Astra Trident。

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

b. 將組態套用至叢集。

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

2. 建立權限群組的身分識別對應。

使用**ekscl**

使用eksctl在叢集與BlueXP Connector的IAM角色之間建立IAM身分識別對應。

"如需完整說明、請參閱eksctl文件"。

以下為範例。

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

編輯**AWS/AUTH**

直接編輯AWS/AUTH ConfigMap、將RBAC存取權限新增至BlueXP Connector的IAM角色。

"如需完整指示、請參閱AWS EKS文件"。

以下為範例。

```
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叢集的需求

您可以使用BluesXP在Azure中新增及管理託管Azure Kubernetes叢集（KS）和自我管理的Kubernetes叢集。在您將叢集新增至BlueXP之前、請先確認符合下列需求。

本主題使用_Kubernetes叢集_、其中的設定與自我管理Kubernetes叢集的組態相同。叢集類型是在組態不同的地方指定。

需求

Astra Trident

需要最新版Astra Trident的四種版本之一。您可以直接從BlueXP安裝或升級Astra Trident。您應該 "檢閱先決條件" 安裝Astra Trident之前。

Cloud Volumes ONTAP

必須將其設定為叢集的後端儲存設備。Cloud Volumes ONTAP "如需組態步驟、請前往Astra Trident文件"。

BlueXP Connector

連接器必須在具備必要權限的Azure中執行。 深入瞭解。

網路連線能力

Kubernetes叢集和Connector之間、以及Kubernetes叢集和Cloud Volumes ONTAP 整個過程之間、都需要網路連線。 深入瞭解。

RBAC授權

BlueXP支援使用及不使用Active Directory的RBAC叢集。每個Azure叢集都必須授權BlueXP Connector角色。 深入瞭解。

準備連接器

Azure中的BlueXP Connector需要探索及管理Kubernetes叢集。您需要建立新的Connector、或是使用具有所需權限的現有Connector。

建立新的**Connector**

請遵循下列其中一個連結中的步驟。

- "從BlueXP建立連接器" （建議）
- "從Azure Marketplace建立連接器"
- "在現有的Linux主機上安裝Connector"

將必要的權限新增至現有的**Connector**（以探索託管的高層叢集）

如果您想要探索託管的高效能叢集、可能需要修改Connector的自訂角色、以提供權限。

步驟

1. 識別指派給Connector虛擬機器的角色：
   a. 在Azure入口網站中、開啟虛擬機器服務。
   b. 選取 Connector 虛擬機器。
   c. 在「設定」下、選取「身分識別」。
   d. 按一下* Azure角色指派*。
   e. 記下指派給Connector虛擬機器的自訂角色。
2. 更新自訂角色：
   a. 在Azure入口網站中、開啟您的Azure訂閱。

b. 按一下*存取控制（IAM）>角色*。

c. 按一下自訂角色的省略符號（...）、然後按一下*編輯*。

d. 按一下Json並新增下列權限：

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

e. 按一下「檢閱+更新」、然後按一下「更新」。

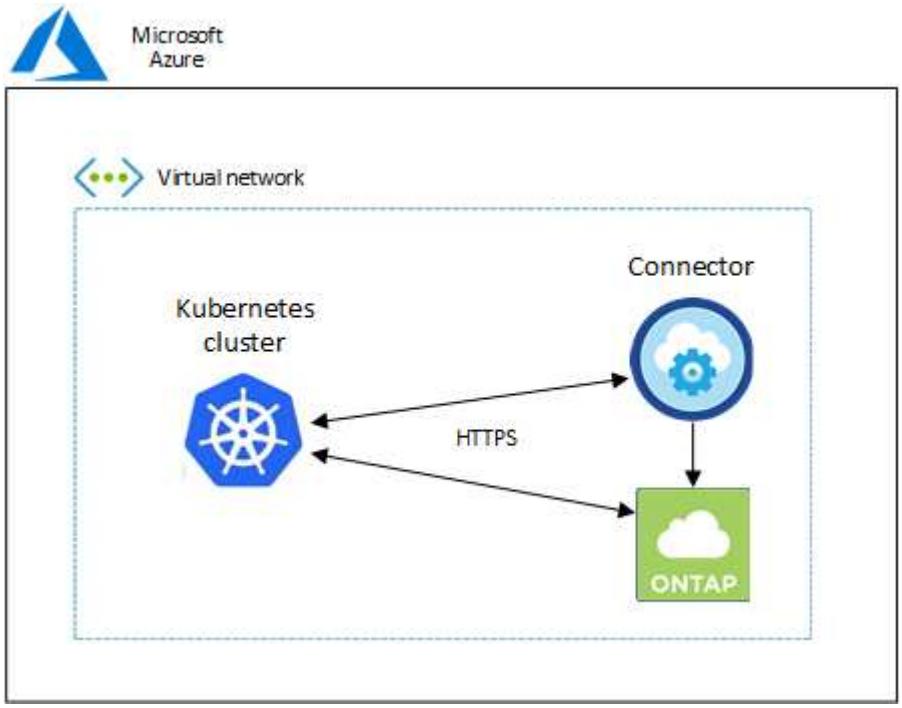
檢閱網路需求

您需要在Kubernetes叢集與Connector之間、以及Kubernetes叢集與Cloud Volumes ONTAP 為叢集提供後端儲存功能的支援系統之間、提供網路連線。

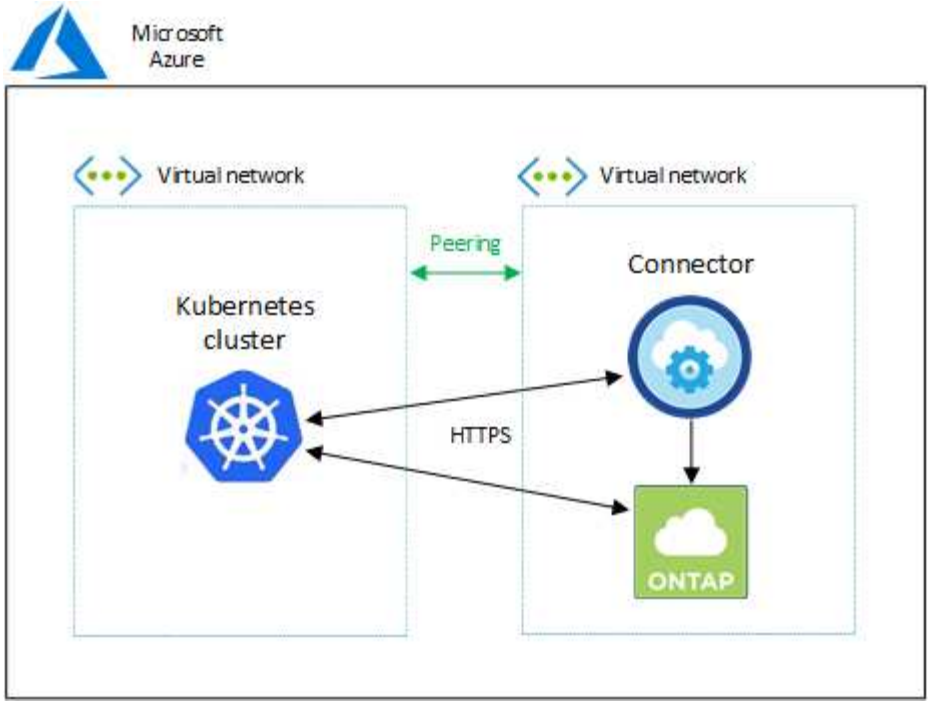
- 每個Kubernetes叢集都必須有來自Connector的傳入連線
- 連接器必須透過連接埠443連線至每個Kubernetes叢集

提供這種連線能力的最簡單方法、就是將Connector和Cloud Volumes ONTAP DB2部署在Kubernetes叢集所在的相同vnet中。否則、您需要在不同的VNets之間設定對等連線。

以下範例顯示同一個vnet中的每個元件。



以下是另一個範例、顯示Kubernetes叢集在不同的vnet上執行。在此範例中、對等功能可在Kubernetes叢集的vnet與Connector和Cloud Volumes ONTAP 物件的vnet之間建立連線。



設定RBAC授權

RBAC驗證只會在啟用Active Directory（AD）的Kubernetes叢集上執行。未使用AD的Kubernetes叢集將自動通過驗證。

您需要在每個Kubernetes叢集上授權Connector角色、以便Connector探索及管理叢集。

備份與還原

備份與還原僅需基本授權。

新增儲存類別

若要使用BlueXP新增儲存類別、並監控叢集是否有變更後端、則需要擴充授權。

安裝**Astra Trident**

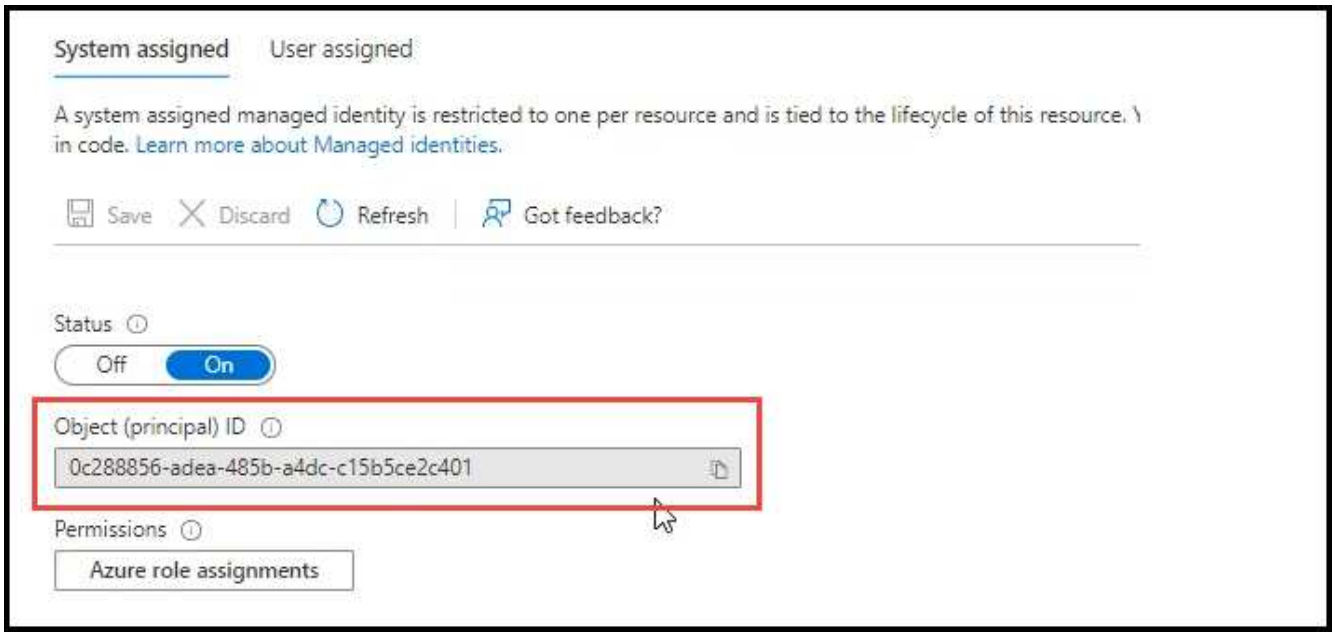
您必須提供BlueXP的完整授權、才能安裝Astra Trident。

> 安裝Astra Trident時、BlueXP會安裝Astra Trident後端和Kubernetes機密、其中包含Astra Trident與儲存叢集通訊所需的認證資料。

開始之前

您的RBAC「子項目：名稱：」組態會因Kubernetes叢集類型而稍有不同。

- 如果要部署*託管的高層叢集*、則需要連接器系統指派的託管身分識別物件ID。此ID可在Azure管理入口網站取得。



- 如果您要部署*自我管理的Kubernetes叢集*、則需要任何授權使用者的使用者名稱。

步驟

建立叢集角色和角色繫結。

1. 您可以根據自己的需求自訂授權。

備份/還原

新增基本授權以啟用Kubernetes叢集的備份與還原。

更換 `subjects: kind:` 使用您的使用者名稱和 `subjects: name:` 使用系統指派託管身分識別的物件 ID 、或上述任何授權使用者的使用者名稱。

```
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
```

儲存類別

新增擴充授權、以使用BlueXP新增儲存類別。

更換 `subjects: kind:` 使用您的使用者名稱和 `subjects: user:` 使用系統指派託管身分識別的物件 ID 、或上述任何授權使用者的使用者名稱。

```
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
```

Trident 安裝

使用命令列提供完整授權、並讓BlueXP安裝Astra Trident。

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

2. 將組態套用至叢集。

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

Google Cloud中Kubernetes叢集的需求

您可以使用BlueXP在Google中新增及管理託管的Google Kubernetes Engine（GKE）叢集和自我管理的Kubernetes叢集。在您將叢集新增至BlueXP之前、請先確認符合下列需求。

> 本主題使用_Kubernetes叢集_、其中GKE和自我管理Kubernetes叢集的組態相同。叢集類型是在組態不同的地方指定。

需求

Astra Trident

需要最新版Astra Trident的四種版本之一。您可以直接從BlueXP安裝或升級Astra Trident。您應該 "檢閱先決條件" 安裝Astra Trident之前

Cloud Volumes ONTAP

在BlueXP中、必須使用與Kubernetes叢集相同的租戶帳戶、工作區和Connector。Cloud Volumes ONTAP "如需組態步驟、請前往Astra Trident文件"。

BlueXP Connector

Connector必須以必要權限在Google中執行。 深入瞭解。

網路連線能力

Kubernetes叢集和Connector之間、以及Kubernetes叢集和Cloud Volumes ONTAP 整個過程之間、都需要網路連線。 深入瞭解。

RBAC授權

BlueXP支援使用及不使用Active Directory的RBAC叢集。每個GKE叢集都必須授權BlueXP Connector角色。深入瞭解。

準備連接器

需要Google的BlueXP Connector來探索及管理Kubernetes叢集。您需要建立新的Connector、或是使用具有所需權限的現有Connector。

建立新的**Connector**

請遵循下列其中一個連結中的步驟。

- "從BlueXP建立連接器" （建議）
- "在現有的Linux主機上安裝Connector"

將必要權限新增至現有的**Connector**（以探索託管**GKE**叢集）

如果您想要探索託管的GKE叢集、可能需要修改Connector的自訂角色、以提供權限。

步驟

1. 在中 "雲端主控台"請移至*角色*頁面。
2. 使用頁面頂端的下拉式清單、選取包含您要編輯之角色的專案或組織。
3. 按一下自訂角色。
4. 按一下*編輯角色*以更新角色的權限。
5. 按一下「新增權限」、將下列新權限新增至角色。

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

6. 按一下「更新」以儲存編輯過的角色。

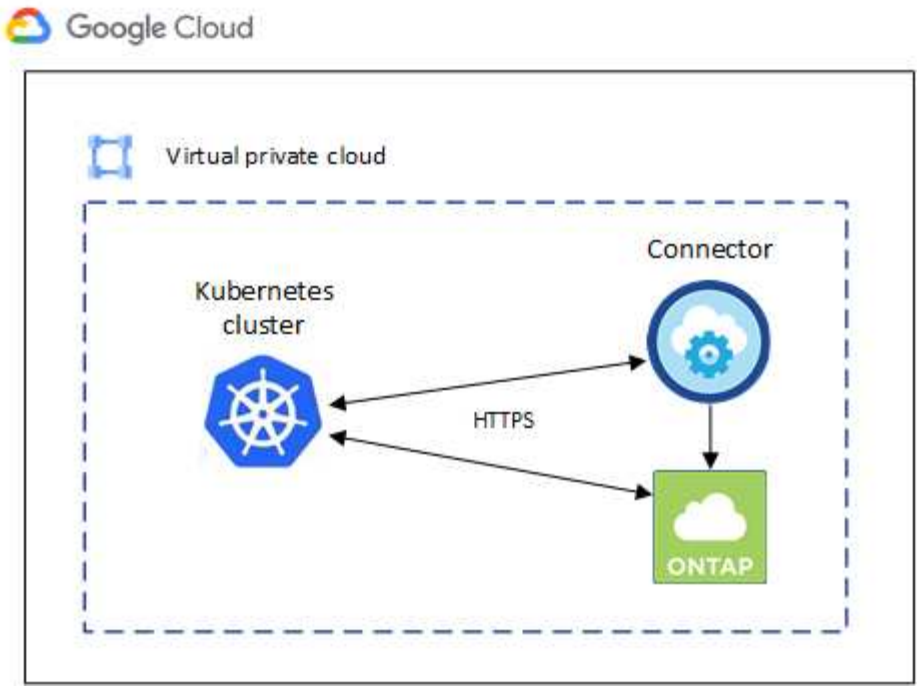
檢閱網路需求

您需要在Kubernetes叢集與Connector之間、以及Kubernetes叢集與Cloud Volumes ONTAP 為叢集提供後端儲存功能的支援系統之間、提供網路連線。

- 每個Kubernetes叢集都必須有來自Connector的傳入連線
- 連接器必須透過連接埠443連線至每個Kubernetes叢集

提供這種連線能力的最簡單方法、就是將Connector和Cloud Volumes ONTAP Sfor部署在Kubernetes叢集所在的VPC上。否則、您需要在不同VPC之間設定對等連線。

以下範例顯示同一VPC中的每個元件。



設定RBAC授權

RBAC驗證只會在啟用Active Directory（AD）的Kubernetes叢集上執行。未使用AD的Kubernetes叢集將自動通過驗證。

您需要在每個Kubernetes叢集上授權Connector角色、以便Connector探索及管理叢集。

備份與還原

備份與還原僅需基本授權。

新增儲存類別

若要使用BlueXP新增儲存類別、並監控叢集是否有變更後端、則需要擴充授權。

安裝**Astra Trident**

您必須提供BlueXP的完整授權、才能安裝Astra Trident。

> 安裝Astra Trident時、BlueXP會安裝Astra Trident後端和Kubernetes機密、其中包含Astra Trident與儲存叢集通訊所需的認證資料。

開始之前

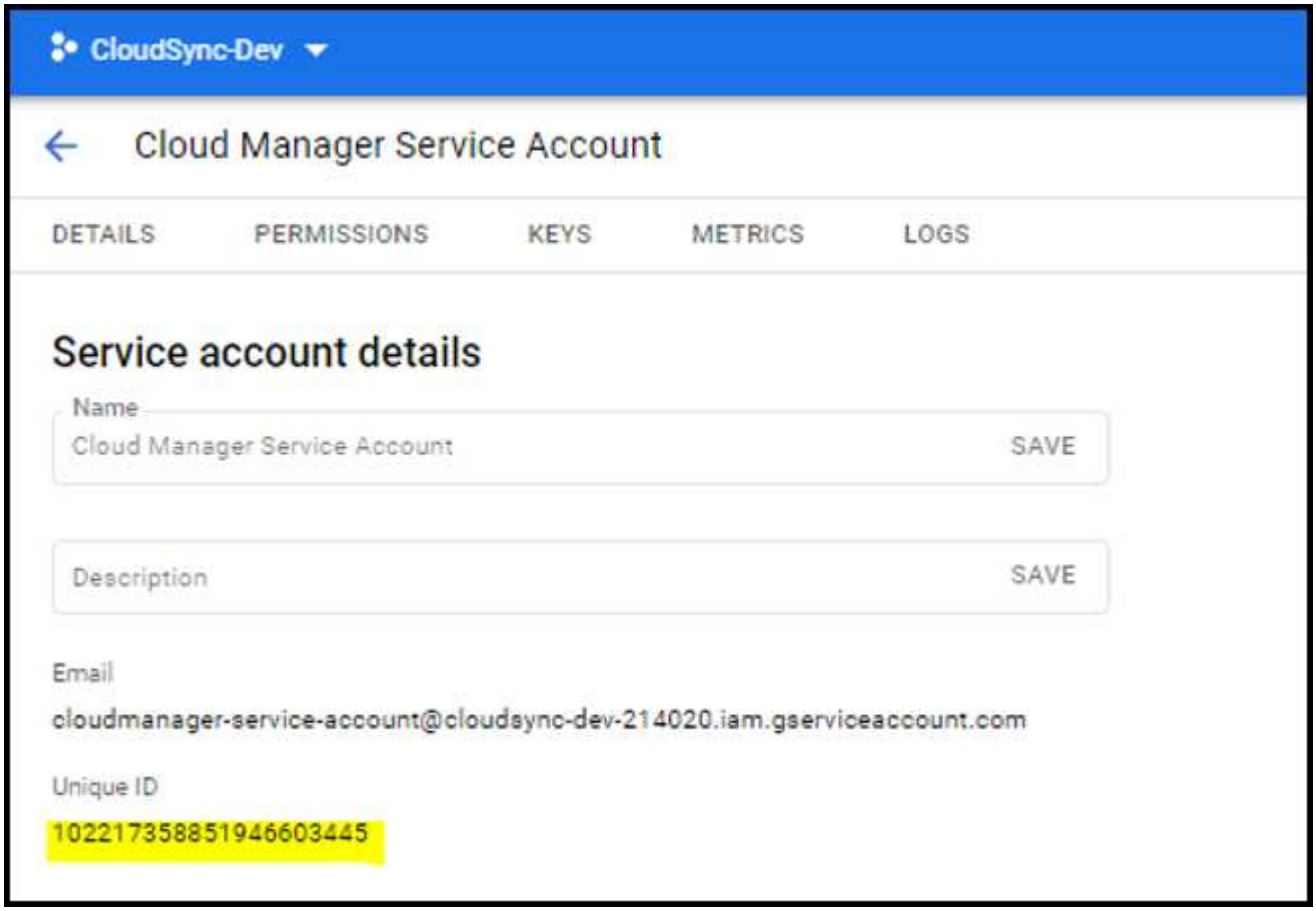
以進行設定 `subjects: name:` 在Yaml檔案中、您需要知道BlueXP的唯一ID。

您可以透過下列兩種方式找到唯一ID：

- 使用命令：

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

- 在的「服務帳戶詳細資料」中 "雲端主控台"。



步驟

建立叢集角色和角色繫結。

1. 您可以根據自己的需求自訂授權。

備份/還原

新增基本授權以啟用Kubernetes叢集的備份與還原。

更換 `subjects: kind:` 使用您的使用者名稱和 `subjects: name:` 具有授權服務帳戶的唯一 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
```

儲存類別

新增擴充授權、以使用BlueXP新增儲存類別。

更換 `subjects: kind:` 使用您的使用者名稱和 `subjects: user:` 具有授權服務帳戶的唯一 ID 。

```
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
```

Trident 安裝

使用命令列提供完整授權、並讓BlueXP安裝Astra Trident。

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

2. 將組態套用至叢集。

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

OpenShift中的Kubernetes叢集需求

您可以使用BlueXP新增及管理自我管理的OpenShift Kubernetes叢集。在您將叢集新增至BlueXP之前、請先確認符合下列需求。

需求

Astra Trident

需要最新版Astra Trident的四種版本之一。您可以直接從BlueXP安裝或升級Astra Trident。您應該 "檢閱先決條件" 安裝Astra Trident之前。

Cloud Volumes ONTAP

必須將其設定為叢集的後端儲存設備。Cloud Volumes ONTAP "如需組態步驟、請前往Astra Trident文件"。

BlueXP Connector

需要使用BlueXP Connector來匯入及管理Kubernetes叢集。您需要建立新的Connector、或是使用具有雲端供應商所需權限的現有Connector：

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

網路連線能力

Kubernetes叢集和Connector之間、以及Kubernetes叢集和Cloud Volumes ONTAP 整個過程之間、都需要網路連線。

具有**RBAC**授權的**Kubernetes**組態檔（**Kubeconfig**）

若要匯入OpenShift叢集、您需要具備RBAC授權的Kbeconfig檔案、才能啟用不同的功能。 建立Kbeconfig檔案。

- 備份與還原：備份與還原僅需基本授權。
- 新增儲存類別：需要擴充授權、才能使用BlueXP新增儲存類別、並監控叢集是否有變更後端。
- 安裝Astra Trident：您需要提供BlueXP的完整授權、才能安裝Astra Trident。

> 安裝Astra Trident時、BlueXP會安裝Astra Trident後端和Kubernetes機密、其中包含Astra Trident與儲存叢集通訊所需的認證資料。

建立Kbeconfig檔案

使用OpenShift CLI、建立要匯入至BlueXP的Kbeconfig檔案。

步驟

1. 使用管理使用者在公共URL上的「ocLogin」（身分登入）登入OpenShift CLI。
2. 建立服務帳戶、如下所示：

   a. 建立名為「oc-service-account.yaml」的服務帳戶檔案。

   視需要調整名稱和命名空間。如果在此處進行變更、您應該在下列步驟中套用相同的變更。

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

+

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

   a. 套用服務帳戶：

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

3. 根據您的授權需求建立自訂角色繫結。

   a. 建立名為「oc-clusterrolebind.yaml」的「ClusterRoeBinding」檔案。

```
oc-clusterrolebinding.yaml
```

   b. 視需要為叢集設定RBAC授權。

備份/還原

新增基本授權以啟用Kubernetes叢集的備份與還原。

```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
```

儲存類別

新增擴充授權、以使用BlueXP新增儲存類別。

```
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
```

Trident 安装

授予完整的系統管理授權、並讓BlueXP能夠安裝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. 套用叢集角色繫結：

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

4. 列出服務帳戶機密、將「<內容>」取代為正確的安裝內容：

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

輸出的結尾應類似於下列內容：

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

"secretts "陣列中每個元素的索引以0開頭。在上述範例中、「oc-service-account-dockercfg-vhz87」的索引為0、而「oc-service-account-toke-r59kr"的索引則為1。在輸出中、記下含有「權杖」一詞的服務帳戶名稱索引。

5. 產生以下的Kbeconfig：

a. 建立「cree-kupeconfig・sh」檔案。將下列指令碼開頭的「toke_index」取代為正確的值。

```
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. 請輸入命令以將其套用至Kubernetes叢集。

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

結果

您將使用所產生的 `kubeconfig-sa` 將OpenShift叢集新增至藍圖XP的檔案。