Requisiti

Kubernetes clusters

NetApp
April 16, 2024

Sommario

Requisiti

Requisiti per i cluster Kubernetes in AWS

Puoi aggiungere cluster Amazon Elastic Kubernetes Service (EKS) gestiti o cluster Kubernetes autogestiti su AWS a BlueXP. Prima di aggiungere i cluster a BlueXP, è necessario assicurarsi che siano soddisfatti i seguenti requisiti.

Questo argomento utilizza *Kubernetes cluster* dove la configurazione è la stessa per i cluster EKS e Kubernetes autogestiti. Viene specificato il tipo di cluster in cui la configurazione differisce.

Requisiti

Astra Trident

È necessaria una delle quattro versioni più recenti di Astra Trident. Puoi installare o aggiornare Astra Trident direttamente da BlueXP. Dovresti "esaminare i prerequisiti" Prima di installare Astra Trident.

Cloud Volumes ONTAP

Cloud Volumes ONTAP per AWS deve essere configurato come storage back-end per il cluster. "Consultare i documenti di Astra Trident per la procedura di configurazione".

Connettore BlueXP

Un connettore deve essere in esecuzione in AWS con le autorizzazioni richieste. Scopri di più di seguito.

Connettività di rete

È necessaria la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e Cloud Volumes ONTAP. Scopri di più di seguito.

Autorizzazione RBAC

Il ruolo BlueXP Connector deve essere autorizzato su ciascun cluster Kubernetes. Scopri di più di seguito.

Preparare un connettore

In AWS è necessario un connettore BlueXP per rilevare e gestire i cluster Kubernetes. Sarà necessario creare un nuovo connettore o utilizzare un connettore esistente con le autorizzazioni richieste.

Creare un nuovo connettore

Seguire la procedura descritta in uno dei collegamenti riportati di seguito.

- "Creare un connettore da BlueXP" (consigliato)
- "Creare un connettore da AWS Marketplace"
- "Installare il connettore su un host Linux esistente in AWS"

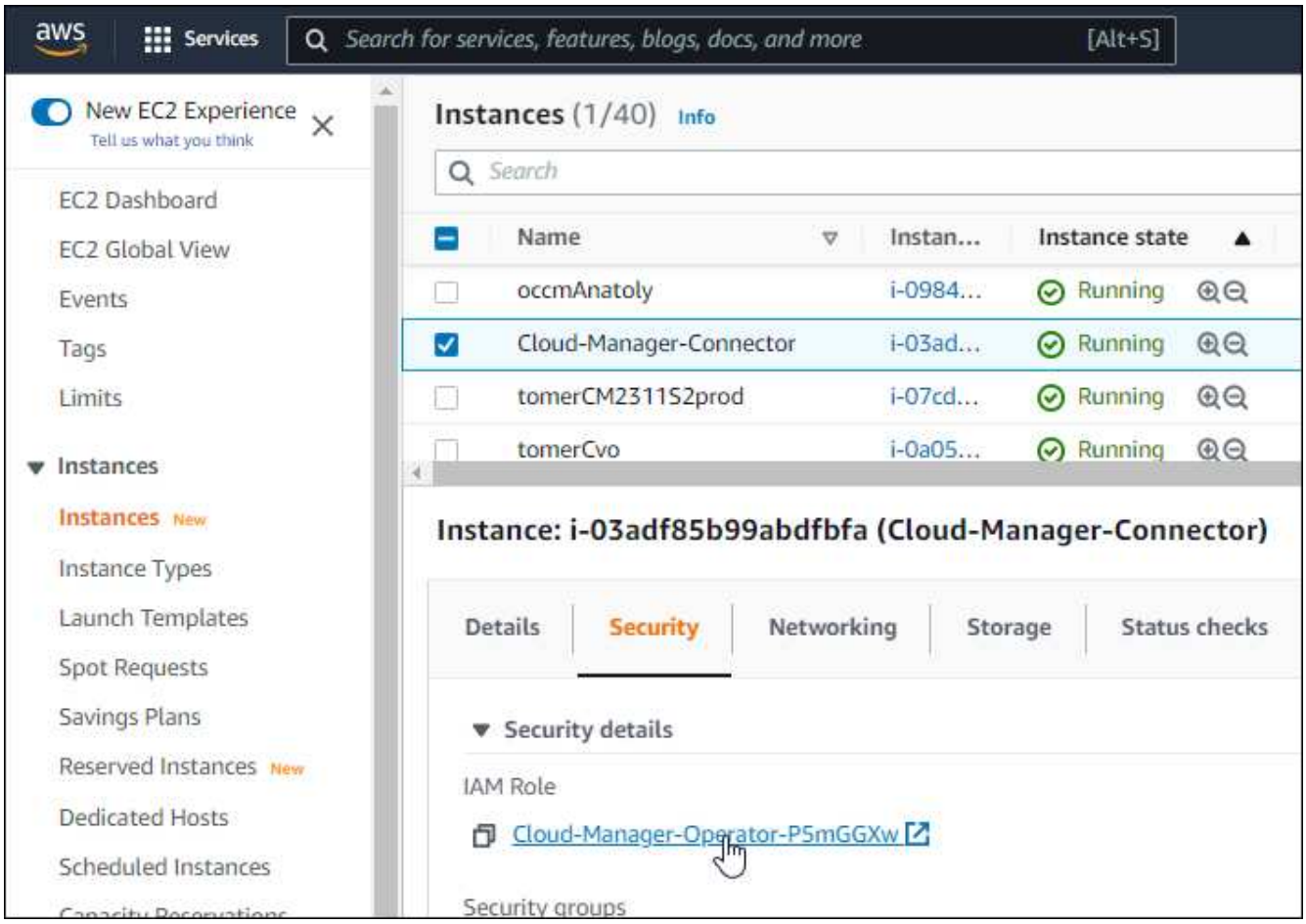
Aggiungere le autorizzazioni richieste a un connettore esistente

A partire dalla release 3.9.13, tutti i _connettori appena creati includono tre nuove autorizzazioni AWS che consentono il rilevamento e la gestione dei cluster Kubernetes. Se è stato creato un connettore prima di

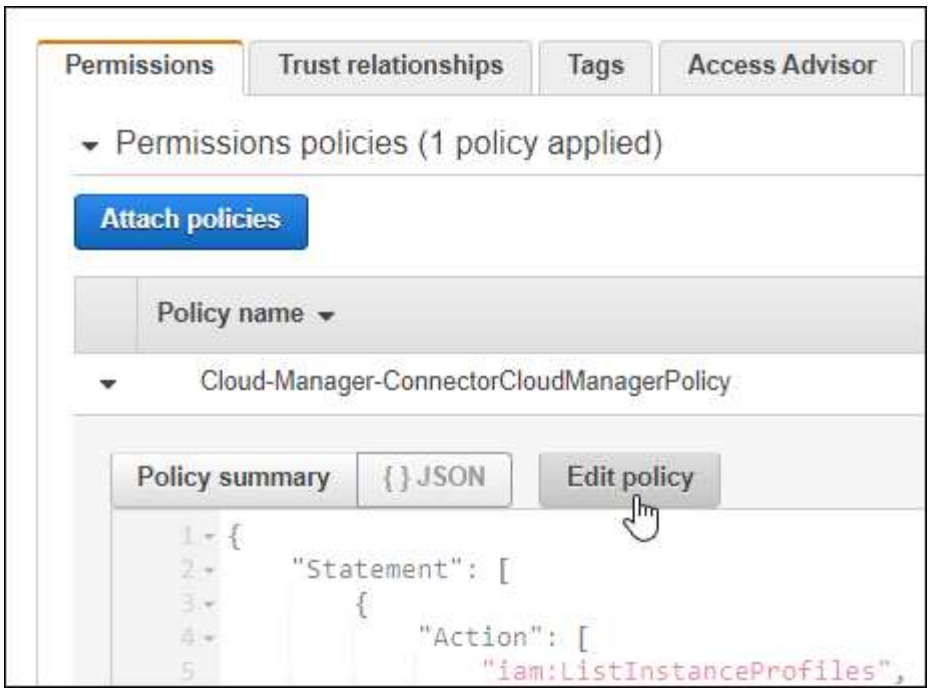
questa release, sarà necessario modificare il criterio esistente per il ruolo IAM del connettore per fornire le autorizzazioni.

Fasi

1. Accedere alla console AWS e aprire il servizio EC2.
2. Selezionare l'istanza del connettore, fare clic su **Security** e fare clic sul nome del ruolo IAM per visualizzare il ruolo nel servizio IAM.



3. Nella scheda **Permissions**, espandere il criterio e fare clic su **Edit policy**.



4. Fare clic su **JSON** e aggiungere le seguenti autorizzazioni nella prima serie di azioni:
   - ec2:DescripteRegions
   - eks:ListClusters
   - eks: DescripbeCluster
   - iam:GetInstanceProfile

   "Visualizza il formato JSON completo per la policy"

5. Fare clic su **Review policy** (esamina policy), quindi su **Save changes** (Salva modifiche).

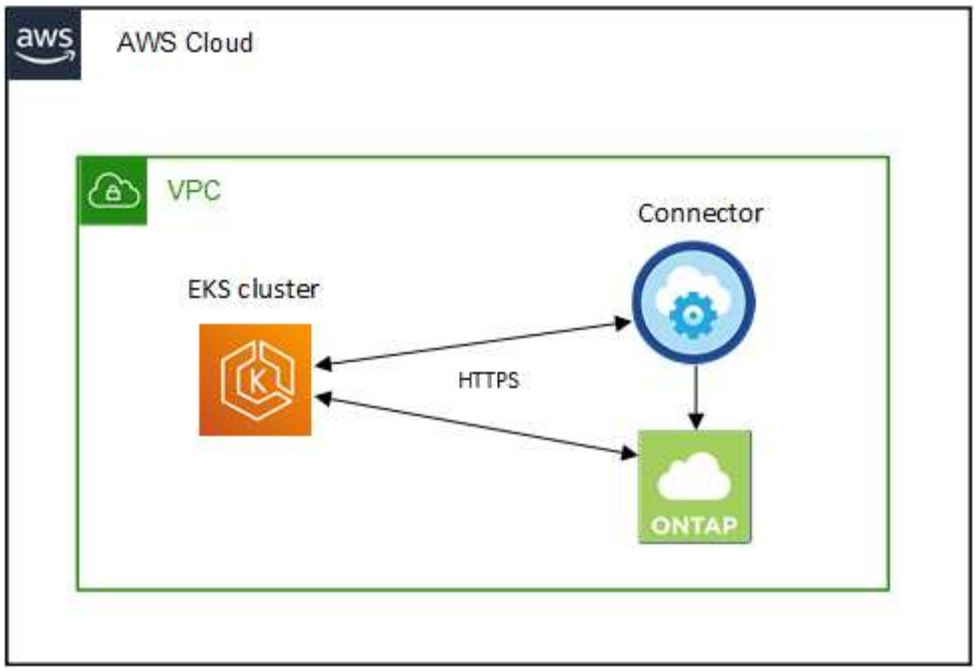
Esaminare i requisiti di rete

È necessario fornire la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e il sistema Cloud Volumes ONTAP che fornisce lo storage back-end al cluster.

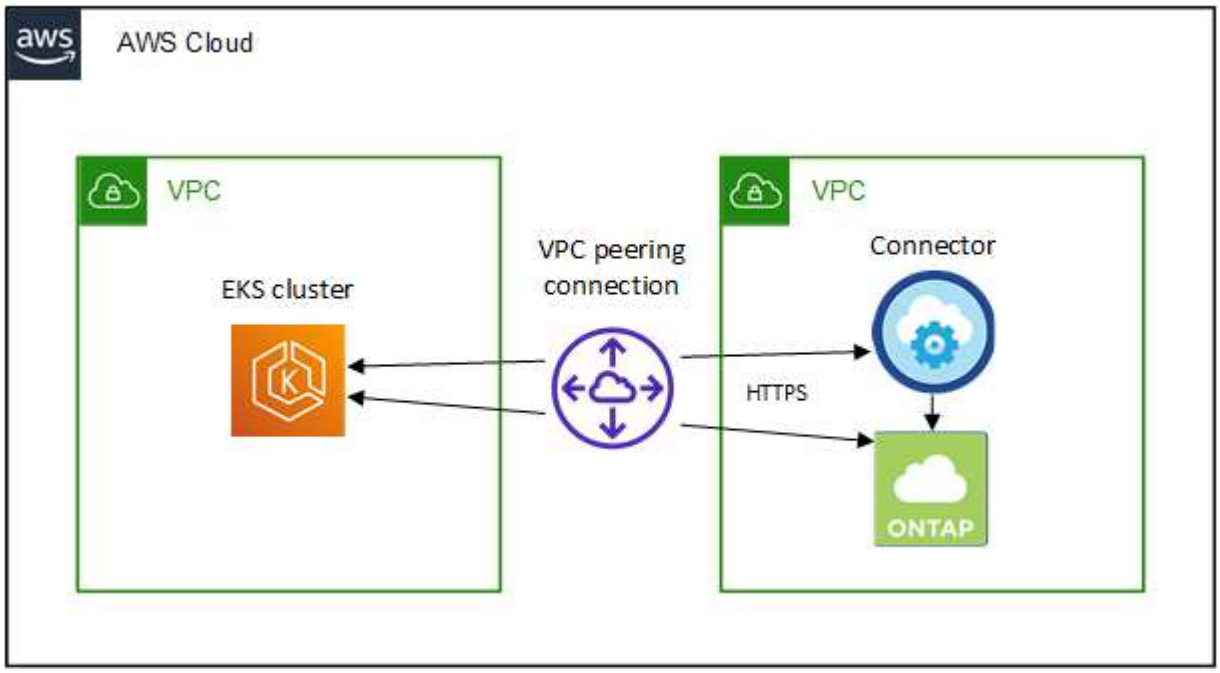
- Ogni cluster Kubernetes deve disporre di una connessione in entrata dal connettore
- Il connettore deve disporre di una connessione in uscita a ciascun cluster Kubernetes sulla porta 443

Il modo più semplice per fornire questa connettività consiste nell'implementare il connettore e Cloud Volumes ONTAP nello stesso VPC del cluster Kubernetes. In caso contrario, è necessario impostare una connessione di peering VPC tra i diversi VPC.

Ecco un esempio che mostra ogni componente dello stesso VPC.



Ecco un altro esempio che mostra un cluster EKS in esecuzione in un VPC diverso. In questo esempio, il peering VPC fornisce una connessione tra il VPC per il cluster EKS e il VPC per il connettore e Cloud Volumes ONTAP.



Impostare l'autorizzazione RBAC

È necessario autorizzare il ruolo del connettore su ciascun cluster Kubernetes in modo che il connettore possa rilevare e gestire un cluster.

Per abilitare funzionalità diverse è necessaria un'autorizzazione diversa.

Backup e ripristino

Il backup e il ripristino richiedono solo un'autorizzazione di base.

Aggiungere classi di storage

È necessaria un'autorizzazione estesa per aggiungere classi di storage utilizzando BlueXP e monitorare il cluster per rilevare eventuali modifiche al backend.

Installare Astra Trident

Devi fornire l'autorizzazione completa per BlueXP per installare Astra Trident.

> Durante l'installazione di Astra Trident, BlueXP installa il backend Astra Trident e il segreto Kubernetes che contiene le credenziali che Astra Trident deve comunicare con il cluster di storage.

Fasi

1. Creare un ruolo del cluster e un'associazione di ruoli.
    a. Puoi personalizzare l'autorizzazione in base ai tuoi requisiti.

Backup/ripristino

Aggiungere l'autorizzazione di base per abilitare il backup e il ripristino per i cluster Kubernetes.

```
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
```

Classi di storage

Aggiunta di autorizzazioni estese per aggiungere classi di storage utilizzando BlueXP.

```
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
```

Installazione di Trident

Utilizzare la riga di comando per fornire l'autorizzazione completa e abilitare BlueXP per installare Astra Trident.

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

b. Applicare la configurazione a un cluster.

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

2. Creare un mapping di identità per il gruppo di autorizzazioni.

Utilizzare eksctl

Utilizzare eksctl per creare una mappatura delle identità IAM tra un cluster e il ruolo IAM per BlueXP Connector.

"Per istruzioni complete, consultare la documentazione eksctl".

Di seguito viene fornito un esempio.

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

Modifica aws-auth

Modificare direttamente aws-auth ConfigMap per aggiungere l'accesso RBAC al ruolo IAM per BlueXP Connector.

"Per istruzioni complete, consultare la documentazione di AWS EKS".

Di seguito viene fornito un esempio.

```
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
```

Requisiti per i cluster Kubernetes in Azure

È possibile aggiungere e gestire cluster Azure Kubernetes gestiti (AKS) e cluster Kubernetes autogestiti in Azure utilizzando BlueXP. Prima di aggiungere i cluster a BlueXP, assicurarsi che siano soddisfatti i seguenti requisiti.

Questo argomento utilizza *Kubernetes cluster* dove la configurazione è la stessa per i cluster AKS e Kubernetes autogestiti. Viene specificato il tipo di cluster in cui la configurazione differisce.

Requisiti

Astra Trident

È necessaria una delle quattro versioni più recenti di Astra Trident. Puoi installare o aggiornare Astra Trident direttamente da BlueXP. Dovresti "esaminare i prerequisiti" Prima di installare Astra Trident.

Cloud Volumes ONTAP

Cloud Volumes ONTAP deve essere configurato come storage back-end per il cluster. "Consultare i documenti di Astra Trident per la procedura di configurazione".

Connettore BlueXP

Un connettore deve essere in esecuzione in Azure con le autorizzazioni richieste. Scopri di più di seguito.

Connettività di rete

È necessaria la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e Cloud Volumes ONTAP. Scopri di più di seguito.

Autorizzazione RBAC

BlueXP supporta cluster abilitati per RBAC con e senza Active Directory. Il ruolo BlueXP Connector deve essere autorizzato su ciascun cluster Azure. Scopri di più di seguito.

Preparare un connettore

Per rilevare e gestire i cluster Kubernetes, è necessario un connettore BlueXP in Azure. Sarà necessario creare un nuovo connettore o utilizzare un connettore esistente con le autorizzazioni richieste.

Creare un nuovo connettore

Seguire la procedura descritta in uno dei collegamenti riportati di seguito.

- "Creare un connettore da BlueXP" (consigliato)
- "Creare un connettore da Azure Marketplace"
- "Installare il connettore su un host Linux esistente"

Aggiungere le autorizzazioni richieste a un connettore esistente (per rilevare un cluster AKS gestito)

Se si desidera rilevare un cluster AKS gestito, potrebbe essere necessario modificare il ruolo personalizzato del connettore per fornire le autorizzazioni.

Fasi

1. Identificare il ruolo assegnato alla macchina virtuale Connector:
   a. Nel portale Azure, aprire il servizio macchine virtuali.
   b. Selezionare la macchina virtuale Connector.
   c. In Impostazioni, selezionare **identità**.
   d. Fare clic su **assegnazioni dei ruoli Azure**.

e. Prendere nota del ruolo personalizzato assegnato alla macchina virtuale del connettore.

2. Aggiornare il ruolo personalizzato:
   a. Nel portale Azure, apri il tuo abbonamento ad Azure.
   b. Fare clic su **controllo di accesso (IAM) > ruoli**.
   c. Fare clic sui puntini di sospensione (…) Per il ruolo personalizzato, quindi fare clic su **Modifica**.
   d. Fare clic su JSON e aggiungere le seguenti autorizzazioni:

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

   e. Fare clic su **Review + update**, quindi su **Update**.

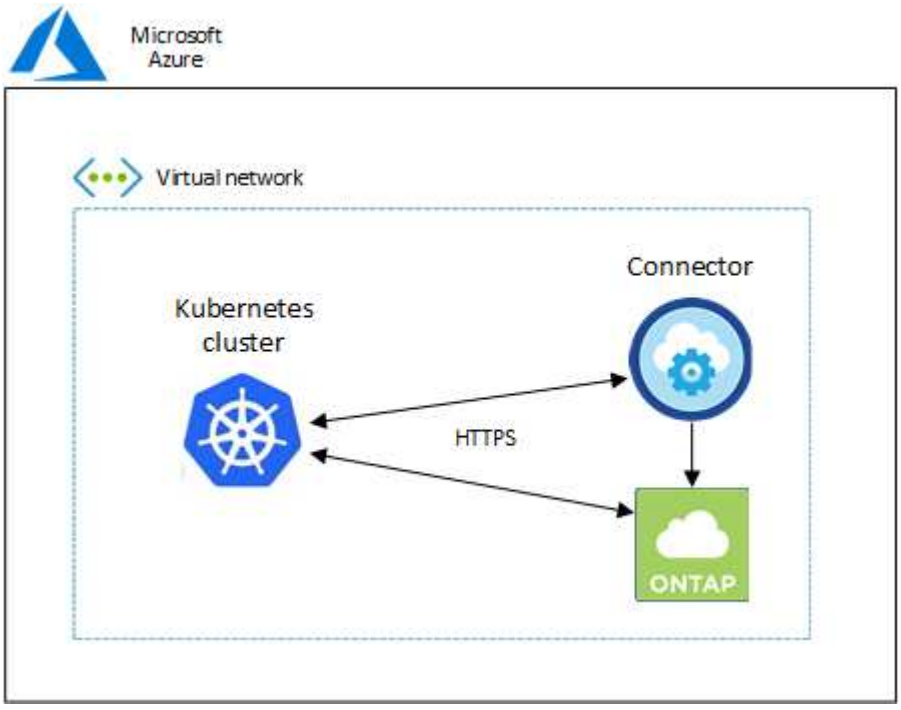
Esaminare i requisiti di rete

È necessario fornire la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e il sistema Cloud Volumes ONTAP che fornisce lo storage back-end al cluster.

- Ogni cluster Kubernetes deve disporre di una connessione in entrata dal connettore
- Il connettore deve disporre di una connessione in uscita a ciascun cluster Kubernetes sulla porta 443

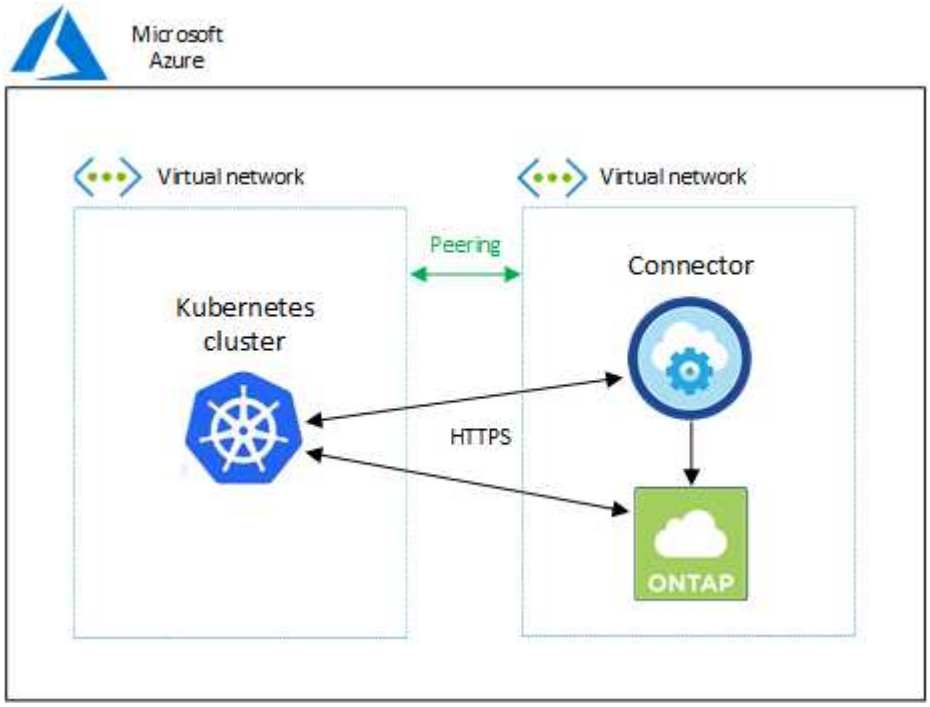
Il modo più semplice per fornire questa connettività consiste nell'implementare il connettore e Cloud Volumes ONTAP nello stesso VNET del cluster Kubernetes. In caso contrario, è necessario impostare una connessione peering tra i diversi VNet.

Di seguito viene riportato un esempio che mostra ciascun componente dello stesso VNET.



Ecco un altro esempio che mostra un cluster Kubernetes in esecuzione in un VNET diverso. In questo

esempio, il peering fornisce una connessione tra VNET per il cluster Kubernetes e VNET per il connettore e Cloud Volumes ONTAP.



Impostare l'autorizzazione RBAC

La convalida RBAC viene eseguita solo sui cluster Kubernetes con Active Directory (ad) attivato. I cluster Kubernetes senza ad passeranno automaticamente la convalida.

È necessario autorizzare il ruolo del connettore su ciascun cluster Kubernetes in modo che il connettore possa rilevare e gestire un cluster.

Backup e ripristino

Il backup e il ripristino richiedono solo un'autorizzazione di base.

Aggiungere classi di storage

È necessaria un'autorizzazione estesa per aggiungere classi di storage utilizzando BlueXP e monitorare il cluster per rilevare eventuali modifiche al backend.

Installare Astra Trident

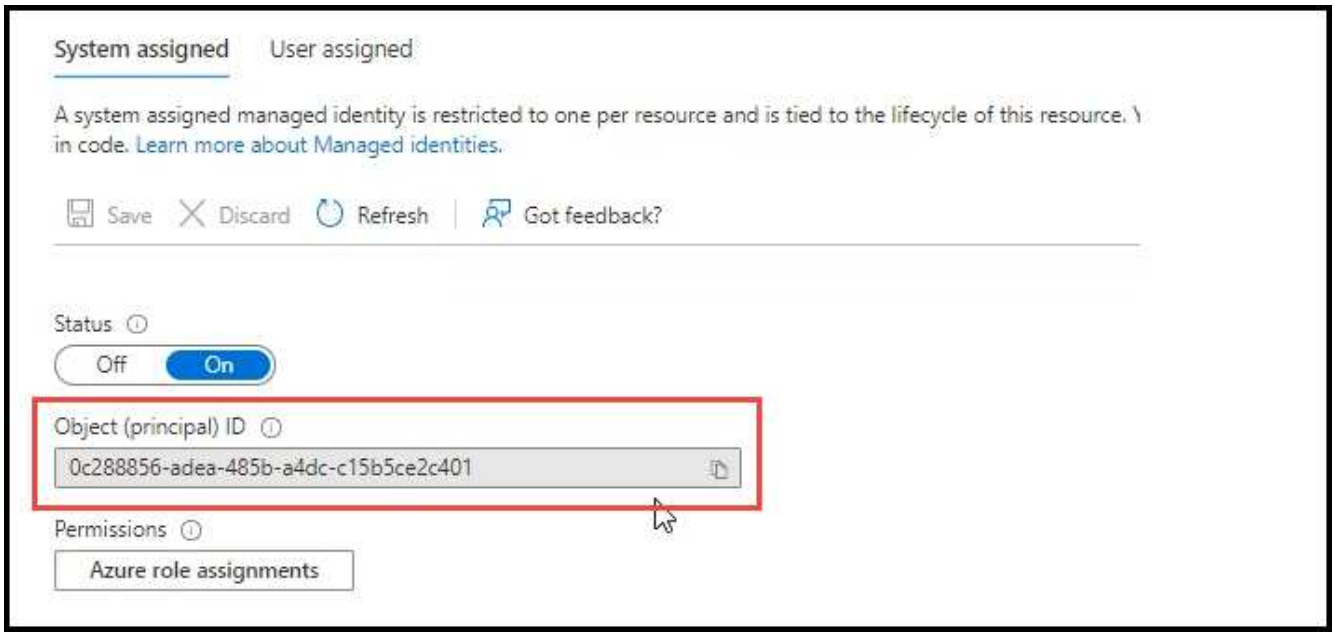
Devi fornire l'autorizzazione completa per BlueXP per installare Astra Trident.

Durante l'installazione di Astra Trident, BlueXP installa il backend Astra Trident e il segreto Kubernetes che contiene le credenziali che Astra Trident deve comunicare con il cluster di storage.

Prima di iniziare

RBAC `subjects: name:` La configurazione varia leggermente in base al tipo di cluster Kubernetes.

- Se si sta implementando un cluster * AKS gestito, è necessario l'ID dell'oggetto per l'identità gestita assegnata dal sistema per il connettore. Questo ID è disponibile nel portale di gestione di Azure.



- Se si sta implementando un cluster Kubernetes* a gestione automatica, è necessario il nome utente di qualsiasi utente autorizzato.

Fasi

Creare un ruolo del cluster e un'associazione di ruoli.

1. Puoi personalizzare l'autorizzazione in base ai tuoi requisiti.

Backup/ripristino

Aggiungere l'autorizzazione di base per abilitare il backup e il ripristino per i cluster Kubernetes.

Sostituire `subjects: kind:` variabile con il nome utente e. `subjects: name:` Con l'ID oggetto per l'identità gestita assegnata dal sistema o il nome utente di qualsiasi utente autorizzato, come descritto in precedenza.

```
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
```

Classi di storage

Aggiunta di autorizzazioni estese per aggiungere classi di storage utilizzando BlueXP.

Sostituire `subjects: kind:` variabile con il nome utente e. `subjects: user:` Con l'ID oggetto per l'identità gestita assegnata dal sistema o il nome utente di qualsiasi utente autorizzato, come descritto in precedenza.

```
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
```

Installazione di Trident

Utilizzare la riga di comando per fornire l'autorizzazione completa e abilitare BlueXP per installare Astra Trident.

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

2. Applicare la configurazione a un cluster.

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

Requisiti per i cluster Kubernetes in Google Cloud

È possibile aggiungere e gestire cluster gestiti di Google Kubernetes Engine (GKE) e cluster di Kubernetes autogestiti in Google utilizzando BlueXP. Prima di aggiungere i cluster a BlueXP, assicurarsi che siano soddisfatti i seguenti requisiti.

In questo argomento viene utilizzato *Kubernetes cluster*, dove la configurazione è la stessa per i cluster GKE e Kubernetes autogestiti. Viene specificato il tipo di cluster in cui la configurazione differisce.

Requisiti

Astra Trident

È necessaria una delle quattro versioni più recenti di Astra Trident. Puoi installare o aggiornare Astra Trident direttamente da BlueXP. Dovresti "esaminare i prerequisiti" Prima di installare Astra Trident

Cloud Volumes ONTAP

Cloud Volumes ONTAP deve essere in BlueXP con lo stesso account di tenancy, spazio di lavoro e connettore del cluster Kubernetes. "Consultare i documenti di Astra Trident per la procedura di configurazione".

Connettore BlueXP

Un connettore deve essere in esecuzione in Google con le autorizzazioni richieste. Scopri di più di seguito.

Connettività di rete

È necessaria la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e

Cloud Volumes ONTAP. Scopri di più di seguito.

Autorizzazione RBAC

BlueXP supporta cluster abilitati per RBAC con e senza Active Directory. Il ruolo BlueXP Connector deve essere autorizzato su ciascun cluster GKE. Scopri di più di seguito.

Preparare un connettore

Per rilevare e gestire i cluster Kubernetes, è necessario un connettore BlueXP in Google. Sarà necessario creare un nuovo connettore o utilizzare un connettore esistente con le autorizzazioni richieste.

Creare un nuovo connettore

Seguire la procedura descritta in uno dei collegamenti riportati di seguito.

- "Creare un connettore da BlueXP" (consigliato)
- "Installare il connettore su un host Linux esistente"

Aggiungere le autorizzazioni richieste a un connettore esistente (per rilevare un cluster GKE gestito)

Se si desidera rilevare un cluster GKE gestito, potrebbe essere necessario modificare il ruolo personalizzato del connettore per fornire le autorizzazioni.

Fasi

1. Poll "Console cloud", Accedere alla pagina **ruoli**.
2. Utilizzando l'elenco a discesa nella parte superiore della pagina, selezionare il progetto o l'organizzazione che contiene il ruolo che si desidera modificare.
3. Fare clic su un ruolo personalizzato.
4. Fare clic su **Edit role** (Modifica ruolo) per aggiornare le autorizzazioni del ruolo.
5. Fare clic su **Add Permissions** (Aggiungi autorizzazioni) per aggiungere le seguenti nuove autorizzazioni al ruolo.

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

6. Fare clic su **Update** (Aggiorna) per salvare il ruolo modificato.

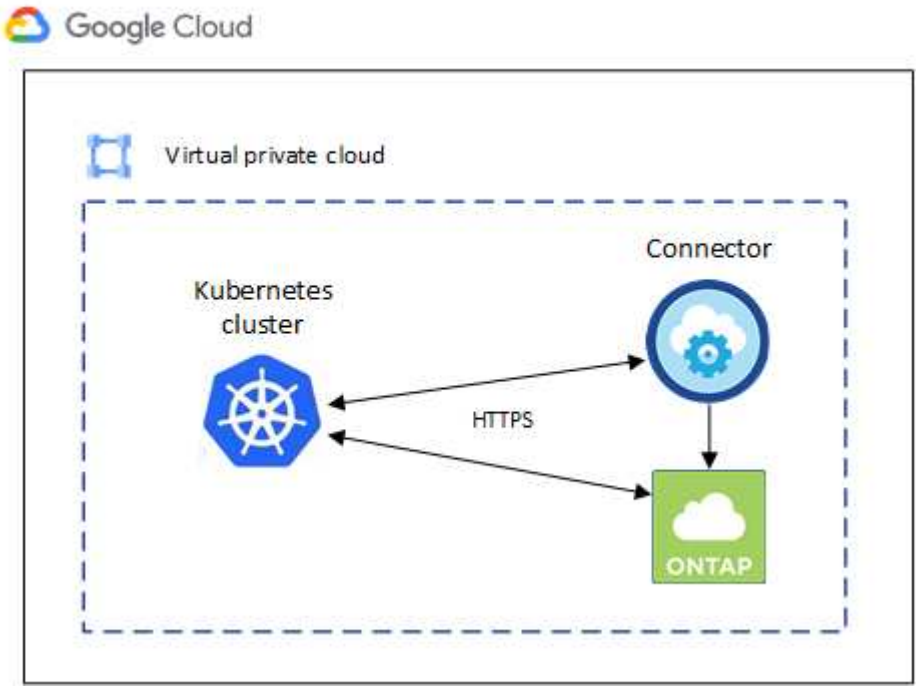
Esaminare i requisiti di rete

È necessario fornire la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e il sistema Cloud Volumes ONTAP che fornisce lo storage back-end al cluster.

- Ogni cluster Kubernetes deve disporre di una connessione in entrata dal connettore
- Il connettore deve disporre di una connessione in uscita a ciascun cluster Kubernetes sulla porta 443

Il modo più semplice per fornire questa connettività consiste nell'implementare il connettore e Cloud Volumes ONTAP nello stesso VPC del cluster Kubernetes. In caso contrario, è necessario impostare una connessione peering tra i diversi VPC.

Ecco un esempio che mostra ogni componente dello stesso VPC.



Impostare l'autorizzazione RBAC

La convalida RBAC viene eseguita solo sui cluster Kubernetes con Active Directory (ad) attivato. I cluster Kubernetes senza ad passeranno automaticamente la convalida.

È necessario autorizzare il ruolo del connettore su ciascun cluster Kubernetes in modo che il connettore possa rilevare e gestire un cluster.

Backup e ripristino

Il backup e il ripristino richiedono solo un'autorizzazione di base.

Aggiungere classi di storage

È necessaria un'autorizzazione estesa per aggiungere classi di storage utilizzando BlueXP e monitorare il cluster per rilevare eventuali modifiche al backend.

Installare Astra Trident

Devi fornire l'autorizzazione completa per BlueXP per installare Astra Trident.

Durante l'installazione di Astra Trident, BlueXP installa il backend Astra Trident e il segreto Kubernetes che contiene le credenziali che Astra Trident deve comunicare con il cluster di storage.

Prima di iniziare

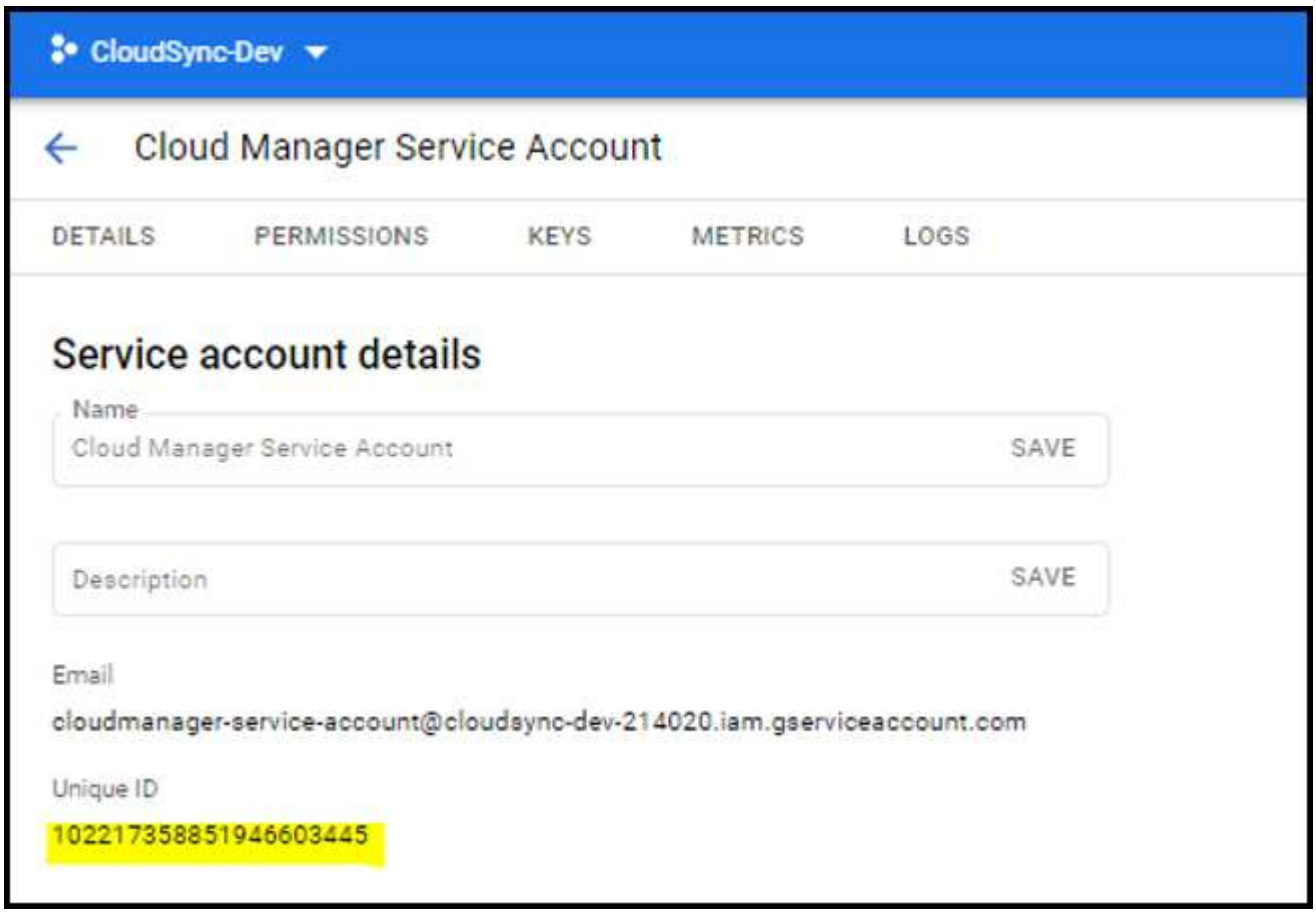
Da configurare `subjects: name:` Nel file YAML, è necessario conoscere l'ID univoco di BlueXP.

Puoi trovare l'ID univoco in due modi:

- Utilizzando il comando:

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

- Nel campo Service account Details (Dettagli account servizio) del "Console cloud".



Fasi

Creare un ruolo del cluster e un'associazione di ruoli.

1. Puoi personalizzare l'autorizzazione in base ai tuoi requisiti.

Backup/ripristino

Aggiungere l'autorizzazione di base per abilitare il backup e il ripristino per i cluster Kubernetes.

Sostituire `subjects: kind:` variabile con il nome utente e. `subjects: name:` Con l'ID univoco dell'account di servizio autorizzato.

```
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
```

Classi di storage

Aggiunta di autorizzazioni estese per aggiungere classi di storage utilizzando BlueXP.

Sostituire `subjects: kind:` variabile con il nome utente e. `subjects: user:` Con l'ID univoco dell'account di servizio autorizzato.

```
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
```

Installazione di Trident

Utilizzare la riga di comando per fornire l'autorizzazione completa e abilitare BlueXP per installare Astra Trident.

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

2. Applicare la configurazione a un cluster.

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

Requisiti per i cluster Kubernetes in OpenShift

È possibile aggiungere e gestire cluster OpenShift Kubernetes autogestiti utilizzando BlueXP. Prima di aggiungere i cluster a BlueXP, assicurarsi che siano soddisfatti i seguenti requisiti.

Requisiti

Astra Trident

È necessaria una delle quattro versioni più recenti di Astra Trident. Puoi installare o aggiornare Astra Trident direttamente da BlueXP. Dovresti "esaminare i prerequisiti" Prima di installare Astra Trident.

Cloud Volumes ONTAP

Cloud Volumes ONTAP deve essere configurato come storage back-end per il cluster. "Consultare i documenti di Astra Trident per la procedura di configurazione".

Connettore BlueXP

Per importare e gestire i cluster Kubernetes è necessario un connettore BlueXP. È necessario creare un nuovo connettore o utilizzare un connettore esistente che disponga delle autorizzazioni necessarie per il provider cloud:

- "Connettore AWS"
- "Connettore Azure"
- "Google Cloud Connector"

Connettività di rete

È necessaria la connettività di rete tra il cluster Kubernetes e il connettore e tra il cluster Kubernetes e Cloud Volumes ONTAP.

File di configurazione di Kubernetes (kubeconfig) con autorizzazione RBAC

Per importare i cluster OpenShift, è necessario un file kubeconfig con l'autorizzazione RBAC richiesta per abilitare funzionalità diverse. Creare un file kubeconfig.

- Backup e ripristino: Il backup e il ripristino richiedono solo un'autorizzazione di base.
- Aggiunta di classi di storage: È richiesta un'autorizzazione estesa per aggiungere classi di storage utilizzando BlueXP e monitorare il cluster per rilevare eventuali modifiche al backend.
- Installare Astra Trident: Devi fornire l'autorizzazione completa per BlueXP per installare Astra Trident.

Durante l'installazione di Astra Trident, BlueXP installa il backend Astra Trident e il segreto Kubernetes che contiene le credenziali che Astra Trident deve comunicare con il cluster di storage.

Creare un file kubeconfig

Utilizzando la CLI di OpenShift, creare un file kubeconfig da importare in BlueXP.

Fasi

1. Accedere alla CLI di OpenShift utilizzando `oc login` Su un URL pubblico con un utente amministrativo.
2. Creare un account di servizio come segue:

   a. Creare un file di account del servizio denominato `oc-service-account.yaml`.

   Regolare il nome e lo spazio dei nomi in base alle esigenze. Se le modifiche vengono apportate qui, è necessario applicare le stesse modifiche nei passaggi seguenti.

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

+

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

   a. Applicare l'account del servizio:

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

3. Creare un'associazione di ruoli personalizzata in base ai requisiti di autorizzazione.

   a. Creare un `ClusterRoleBinding` file chiamato `oc-clusterrolebinding.yaml`.

```
oc-clusterrolebinding.yaml
```

b. Configurare l’autorizzazione RBAC in base alle esigenze del cluster.

Backup/ripristino

Aggiungere l'autorizzazione di base per abilitare il backup e il ripristino per i cluster Kubernetes.

```
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
```

Classi di storage

Aggiunta di autorizzazioni estese per aggiungere classi di storage utilizzando BlueXP.

```
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
```

Installazione di Trident

Concedere l'autorizzazione amministrativa completa e abilitare BlueXP per l'installazione di 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. Applicare l'associazione del ruolo del cluster:

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

4. Elencare i segreti dell'account di servizio, sostituendo `<context>` con il contesto corretto per l'installazione:

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

La fine dell'output dovrebbe essere simile a quanto segue:

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

Gli indici di ciascun elemento in `secrets` l'array inizia con 0. Nell'esempio precedente, l'indice per `oc-service-account-dockercfg-vhz87` sarebbe 0 e l'indice per `oc-service-account-token-r59kr` sarebbe 1. Nell'output, annotare l'indice del nome dell'account del servizio che contiene la parola "token".

5. Generare il kubeconfig come segue:

a. Creare un `create-kubeconfig.sh` file. Sostituire `TOKEN_INDEX` all'inizio del seguente script con il valore corretto.

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. Eseguire la sorgente dei comandi per applicarli al cluster Kubernetes.

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

Risultato

Verrà utilizzato il risultato `kubeconfig-sa` File per aggiungere un cluster OpenShift a BlueXP.

Informazioni sul copyright

Informazioni sul marchio commerciale