



Requirements

Kubernetes clusters

NetApp
April 12, 2022

This PDF was generated from <https://docs.netapp.com/us-en/cloud-manager-kubernetes/requirements/kubernetes-reqs-aws.html> on April 12, 2022. Always check docs.netapp.com for the latest.

Table of Contents

- Requirements 1
 - Requirements for Kubernetes clusters in AWS 1
 - Requirements for Kubernetes clusters in Azure 9
 - Requirements for Kubernetes clusters in Google Cloud 16

Requirements

Requirements for Kubernetes clusters in AWS

You can add managed Amazon Elastic Kubernetes Service (EKS) clusters or self-managed Kubernetes clusters on AWS to Cloud Manager. Before you can add the clusters to Cloud Manager, you need to ensure that the following requirements are met.



This topic uses *Kubernetes cluster* where configuration is the same for EKS and self-managed Kubernetes clusters. The cluster type is specified where configuration differs.

Requirements

Astra Trident

One of the four most recent versions of Astra Trident is required. You can install Astra Trident directly from Cloud Manager. You should [review the prerequisites](#) prior to installing Astra Trident.

To upgrade Astra Trident, [upgrade with the operator](#).

Cloud Volumes ONTAP

Cloud Volumes ONTAP for AWS must be set up as backend storage for the cluster. [Go to the Astra Trident docs for configuration steps](#).

Cloud Manager Connector

A Connector must be running in AWS with the required permissions. [Learn more below](#).

Network connectivity

Network connectivity is required between the Kubernetes cluster and the Connector and between the Kubernetes cluster and Cloud Volumes ONTAP. [Learn more below](#).

RBAC authorization

The Cloud Manager Connector role must be authorized on each Kubernetes cluster. [Learn more below](#).

Prepare a Connector

A Cloud Manager Connector is required in AWS to discover and manage Kubernetes clusters. You'll need to create a new Connector or use an existing Connector that has the required permissions.

Create a new Connector

Follow the steps in one of the links below.

- [Create a Connector from Cloud Manager](#) (recommended)
- [Create a Connector from the AWS Marketplace](#)
- [Install the Connector on an existing Linux host in AWS](#)

Add the required permissions to an existing Connector

Starting in the 3.9.13 release, any *newly* created Connectors include three new AWS permissions that enable discovery and management of Kubernetes clusters. If you created a Connector prior to this release, then you'll

need to modify the existing policy for the Connector's IAM role to provide the permissions.

Steps

1. Go the AWS console and open the EC2 service.
2. Select the Connector instance, click **Security**, and click the name of the IAM role to view the role in the IAM service.



3. In the **Permissions** tab, expand the policy and click **Edit policy**.



4. Click **JSON** and add the following permissions under the first set of actions:

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

[View the full JSON format for the policy.](#)

5. Click **Review policy** and then click **Save changes**.

Review networking requirements

You need to provide network connectivity between the Kubernetes cluster and the Connector and between the Kubernetes cluster and the Cloud Volumes ONTAP system that provides backend storage to the cluster.

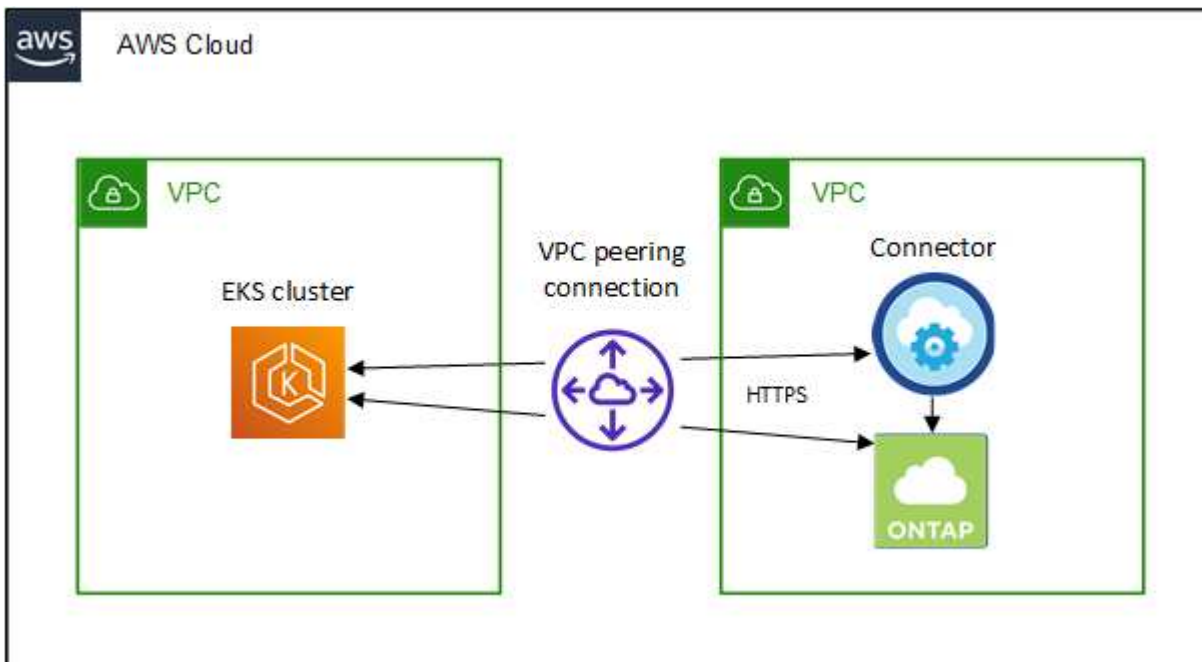
- Each Kubernetes cluster must have an inbound connection from the Connector
- The Connector must have an outbound connection to each Kubernetes cluster over port 443

The simplest way to provide this connectivity is to deploy the Connector and Cloud Volumes ONTAP in the same VPC as the Kubernetes cluster. Otherwise, you need to set up a VPC peering connection between the different VPCs.

Here's an example that shows each component in the same VPC.



And here's another example that shows an EKS cluster running in a different VPC. In this example, VPC peering provides a connection between the VPC for the EKS cluster and the VPC for the Connector and Cloud Volumes ONTAP.



Set up RBAC authorization

You need to authorize the Connector role on each Kubernetes cluster so the Connector can discover and manage a cluster.

Different authorization is required to enable different functionality.

Backup and restore

Backup and restore requires only basic authorization.

Add storage classes

Expanded authorization is required to add storage classes using Cloud Manager.

Install Astra trident

You need to provide full authorization for Cloud Manager to install Astra Trident.



When installing Astra Trident, Cloud Manager installs the Astra Trident backend and Kubernetes secret that contains the credentials Astra Trident needs to communicate with the storage cluster.

Steps

1. Create a cluster role and role binding.
 - a. Create a YAML file that includes the following text based on your authorization requirements.

Backup/restore

Add basic authorization to enable backup and restore for Kubernetes clusters.

```
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

```

Storage classes

Add expanded authorization to add storage classes using Cloud Manager.

```

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

```

Install Trident

Use the command line to provide full authorization and enable Cloud Manager to install Astra Trident.

```

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

```

b. Apply the configuration to a cluster.

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

2. Create an identity mapping to the permissions group using `eksctl`. An example is provided below.

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

[Go to the eksctl documentation for full instructions.](#)

Requirements for Kubernetes clusters in Azure

You can add and manage managed Azure Kubernetes clusters (AKS) and self-managed Kubernetes clusters in Azure using Cloud Manager. Before you can add the clusters to Cloud Manager, ensure the following requirements are met.



This topic uses *Kubernetes cluster* where configuration is the same for AKS and self-managed Kubernetes clusters. The cluster type is specified where configuration differs.

Requirements

Astra Trident

One of the four most recent versions of Astra Trident is required. You can install Astra Trident directly from Cloud Manager. You should [review the prerequisites](#) prior to installing Astra Trident.

To upgrade Astra Trident, [upgrade with the operator](#).

Cloud Volumes ONTAP

Cloud Volumes ONTAP must be set up as backend storage for the cluster. [Go to the Astra Trident docs for configuration steps](#).

Cloud Manager Connector

A Connector must be running in Azure with the required permissions. [Learn more below](#).

Network connectivity

Network connectivity is required between the Kubernetes cluster and the Connector and between the Kubernetes cluster and Cloud Volumes ONTAP. [Learn more below](#).

RBAC authorization

Cloud Manager supports RBAC-enabled clusters with and without Active Directory. The Cloud Manager Connector role must be authorized on each Azure cluster. [Learn more below](#).

Prepare a Connector

A Cloud Manager Connector in Azure is required to discover and manage Kubernetes clusters. You'll need to create a new Connector or use an existing Connector that has the required permissions.

Create a new Connector

Follow the steps in one of the links below.

- [Create a Connector from Cloud Manager](#) (recommended)
- [Create a Connector from the Azure Marketplace](#)
- [Install the Connector on an existing Linux host](#)

Add the required permissions to an existing Connector (to discover a managed AKS cluster)

If you want to discover a managed AKS cluster, you might need to modify the custom role for the Connector to provide the permissions.

Steps

1. Identify the role assigned to the Connector virtual machine:
 - a. In the Azure portal, open the Virtual machines service.
 - b. Select the Connector virtual machine.
 - c. Under Settings, select **Identity**.
 - d. Click **Azure role assignments**.
 - e. Make note of the custom role assigned to the Connector virtual machine.
2. Update the custom role:
 - a. In the Azure portal, open your Azure subscription.
 - b. Click **Access control (IAM) > Roles**.
 - c. Click the ellipsis (...) for the custom role and then click **Edit**.
 - d. Click JSON and add the following permissions:

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

- e. Click **Review + update** and then click **Update**.

Review networking requirements

You need to provide network connectivity between the Kubernetes cluster and the Connector and between the Kubernetes cluster and the Cloud Volumes ONTAP system that provides backend storage to the cluster.

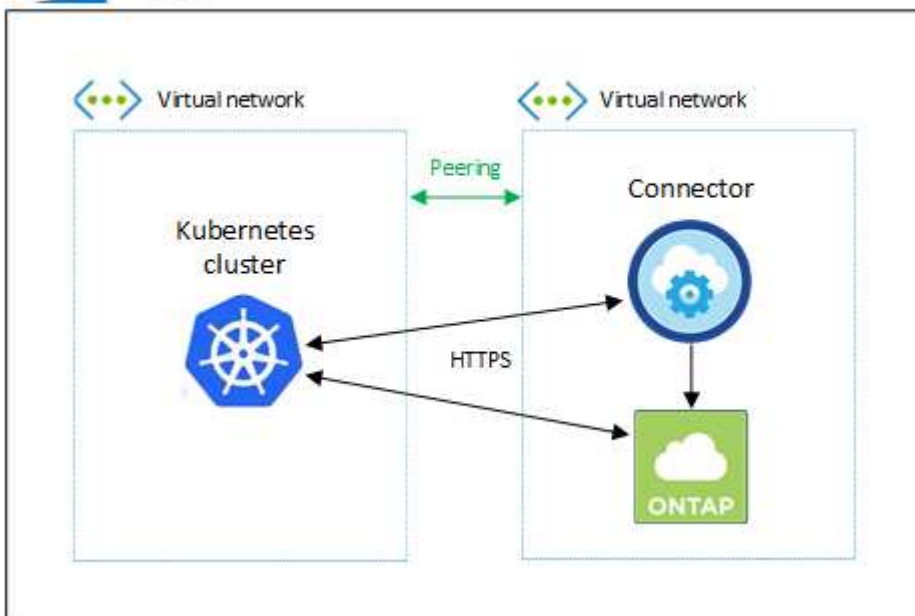
- Each Kubernetes cluster must have an inbound connection from the Connector
- The Connector must have an outbound connection to each Kubernetes cluster over port 443

The simplest way to provide this connectivity is to deploy the Connector and Cloud Volumes ONTAP in the same VNet as the Kubernetes cluster. Otherwise, you need to set up a peering connection between the different VNets.

Here's an example that shows each component in the same VNet.



And here's another example that shows a Kubernetes cluster running in a different VNet. In this example, peering provides a connection between the VNet for the Kubernetes cluster and the VNet for the Connector and Cloud Volumes ONTAP.



Set up RBAC authorization

RBAC validation occurs only on Kubernetes clusters with Active Directory (AD) enabled. Kubernetes clusters without AD will pass validation automatically.

You need authorize the Connector role on each Kubernetes cluster so the Connector can discover and manage a cluster.

Backup and restore

Backup and restore requires only basic authorization.

Add storage classes

Expanded authorization is required to add storage classes using Cloud Manager.

Install Astra trident

You need to provide full authorization for Cloud Manager to install Astra Trident.



When installing Astra Trident, Cloud Manager installs the Astra Trident backend and Kubernetes secret that contains the credentials Astra Trident needs to communicate with the storage cluster.

Before you begin

Your RBAC `subjects: name:` configuration varies slightly based on your Kubernetes cluster type.

- If you are deploying a **managed AKS cluster**, you need the Object ID for the system-assigned managed identity for the Connector. This ID is available in Azure management portal.

The screenshot shows the Azure portal interface for a system-assigned managed identity. The 'System assigned' tab is selected. Below the header, there is a description: 'A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. \n in code. [Learn more about Managed identities.](#)'. Below this, there are buttons for 'Save', 'Discard', 'Refresh', and 'Got feedback?'. The 'Status' is set to 'On'. The 'Object (principal) ID' field is highlighted with a red box, showing the value '0c28856-adea-485b-a4dc-c15b5ce2c401'. Below this, there is a 'Permissions' section with a button for 'Azure role assignments'.

- If you are deploying a **self-managed Kubernetes cluster**, you need the username of any authorized user.

Steps

Create a cluster role and role binding.

1. Create a YAML file that includes the following text based on your authorization requirements. Replace the `subjects: kind: variable` with your username and `subjects: user:` with either the Object ID for the system-assigned managed identity or username of any authorized user as described above.

Backup/restore

Add basic authorization to enable backup and restore for Kubernetes clusters.

```
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

```

Storage classes

Add expanded authorization to add storage classes using Cloud Manager.

```

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

```

Install Trident

Use the command line to provide full authorization and enable Cloud Manager to install Astra Trident.

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

2. Apply the configuration to a cluster.

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

Requirements for Kubernetes clusters in Google Cloud

You can add and manage managed Google Kubernetes Engine (GKE) clusters and self-managed Kubernetes clusters in Google using Cloud Manager. Before you can add the clusters to Cloud Manager, ensure the following requirements are met.



This topic uses *Kubernetes cluster* where configuration is the same for GKE and self-managed Kubernetes clusters. The cluster type is specified where configuration differs.

Requirements

Astra Trident

One of the four most recent versions of Astra Trident is required. You can install Astra Trident directly from Cloud Manager. You should [review the prerequisites](#) prior to installing Astra Trident

To upgrade Astra Trident, [upgrade with the operator](#).

Cloud Volumes ONTAP

Cloud Volumes ONTAP must be in Cloud Manager under the same tenancy account, workspace, and Connector as the Kubernetes cluster. [Go to the Astra Trident docs for configuration steps](#).

Cloud Manager Connector

A Connector must be running in Google with the required permissions. [Learn more below](#).

Network connectivity

Network connectivity is required between the Kubernetes cluster and the Connector and between the Kubernetes cluster and Cloud Volumes ONTAP. [Learn more below](#).

RBAC authorization

Cloud Manager supports RBAC-enabled clusters with and without Active Directory. The Cloud Manager Connector role must be authorized on each GKE cluster. [Learn more below](#).

Prepare a Connector

A Cloud Manager Connector in Google is required to discover and manage Kubernetes clusters. You'll need to create a new Connector or use an existing Connector that has the required permissions.

Create a new Connector

Follow the steps in one of the links below.

- [Create a Connector from Cloud Manager](#) (recommended)
- [Install the Connector on an existing Linux host](#)

Add the required permissions to an existing Connector (to discover a managed GKE cluster)

If you want to discover a managed GKE cluster, you might need to modify the custom role for the Connector to provide the permissions.

Steps

1. In [Cloud Console](#), go to the **Roles** page.

2. Using the drop-down list at the top of the page, select the project or organization that contains the role that you want to edit.
3. Click a custom role.
4. Click **Edit Role** to update the role's permissions.
5. Click **Add Permissions** to add the following new permissions to the role.

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

6. Click **Update** to save the edited role.

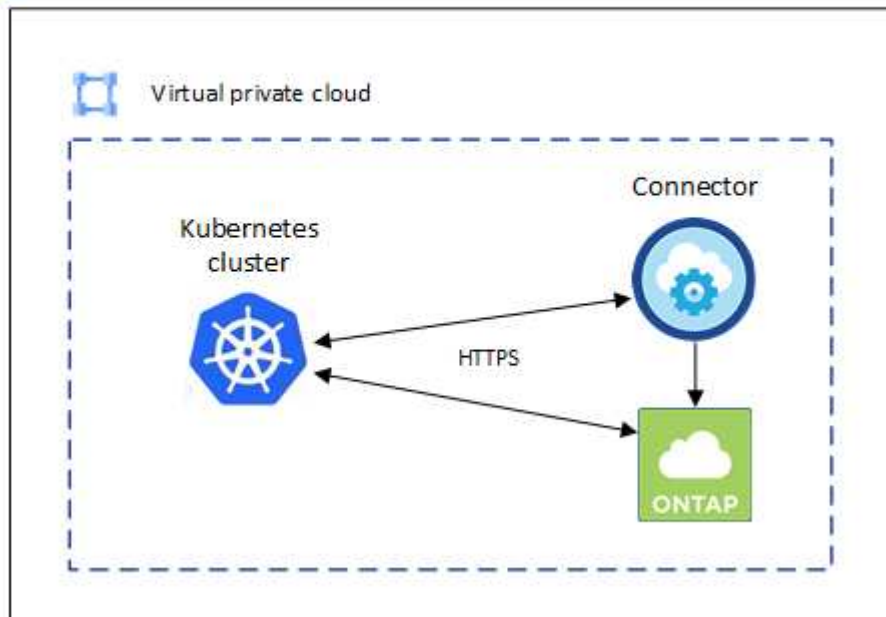
Review networking requirements

You need to provide network connectivity between the Kubernetes cluster and the Connector and between the Kubernetes cluster and the Cloud Volumes ONTAP system that provides backend storage to the cluster.

- Each Kubernetes cluster must have an inbound connection from the Connector
- The Connector must have an outbound connection to each Kubernetes cluster over port 443

The simplest way to provide this connectivity is to deploy the Connector and Cloud Volumes ONTAP in the same VPC as the Kubernetes cluster. Otherwise, you need to set up a peering connection between the different VPC.

Here's an example that shows each component in the same VPC.



Set up RBAC authorization

RBAC validation occurs only on Kubernetes clusters with Active Directory (AD) enabled. Kubernetes clusters without AD will pass validation automatically.

You need authorize the Connector role on each Kubernetes cluster so the Connector can discover and manage a cluster.

Backup and restore

Backup and restore requires only basic authorization.

Add storage classes

Expanded authorization is required to add storage classes using Cloud Manager.

Install Astra trident

You need to provide full authorization for Cloud Manager to install Astra Trident.



When installing Astra Trident, Cloud Manager installs the Astra Trident backend and Kubernetes secret that contains the credentials Astra Trident needs to communicate with the storage cluster.

Before you begin

To configure `subjects: name:` in the YAML file, you need to know the Cloud Manager Unique ID.

You can find the unique ID one of two ways:

- Using the command:

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

- In the Service Account Details on the [Cloud Console](#).

CloudSync-Dev

←

Cloud Manager Service Account

DETAILSPERMISSIONSKEYSMETRICSLOGS

Service account details

Name

Cloud Manager Service Account

SAVE

Description

SAVE

Email

cloudmanager-service-account@cloudsync-dev-214020.iam.gserviceaccount.com

Unique ID

102217358851946603445

Steps

Create a cluster role and role binding.

1. Create a YAML file that includes the following text based on your authorization requirements. Replace the `subjects: kind: variable` with your username and `subjects: user:` with the unique ID for the authorized service account.

Backup/restore

Add basic authorization to enable backup and restore for Kubernetes clusters.

```
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

```

Storage classes

Add expanded authorization to add storage classes using Cloud Manager.

```

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

```

Install Trident

Use the command line to provide full authorization and enable Cloud Manager to install Astra Trident.

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

2. Apply the configuration to a cluster.

```
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.