Get started with Kubernetes clusters in Azure

Cloud Manager

NetApp
February 23, 2022

Table of Contents

Get started with Kubernetes clusters in Azure

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

The Kubernetes cluster must have NetApp Astra Trident deployed. Install one of the four most recent versions of Astra Trident using Helm. Go to the Astra Trident docs for installation steps using Helm.

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**.

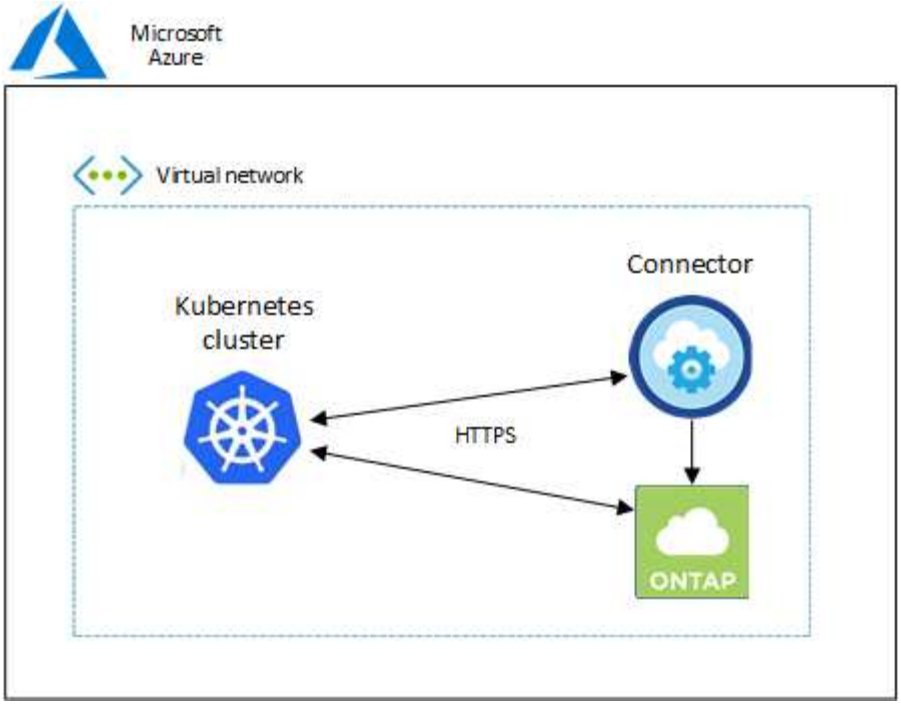
Review networking requirements

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.

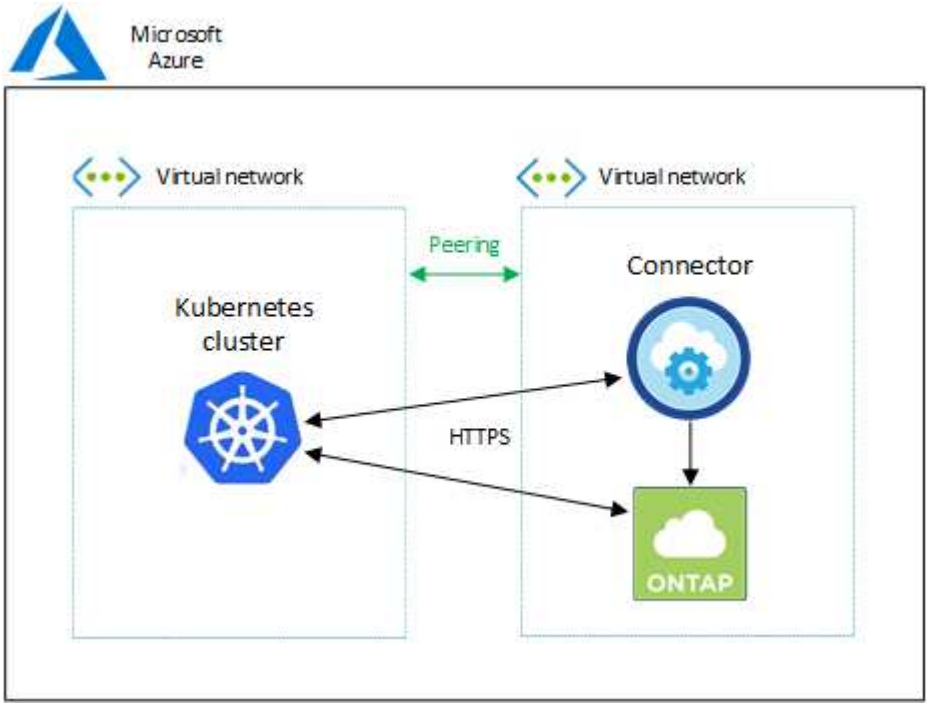
- Each Kubernetes cluster must have an inbound connection from the Connector
- The Connector must have an outbound connection to each Kubernetes cluster over port 443

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

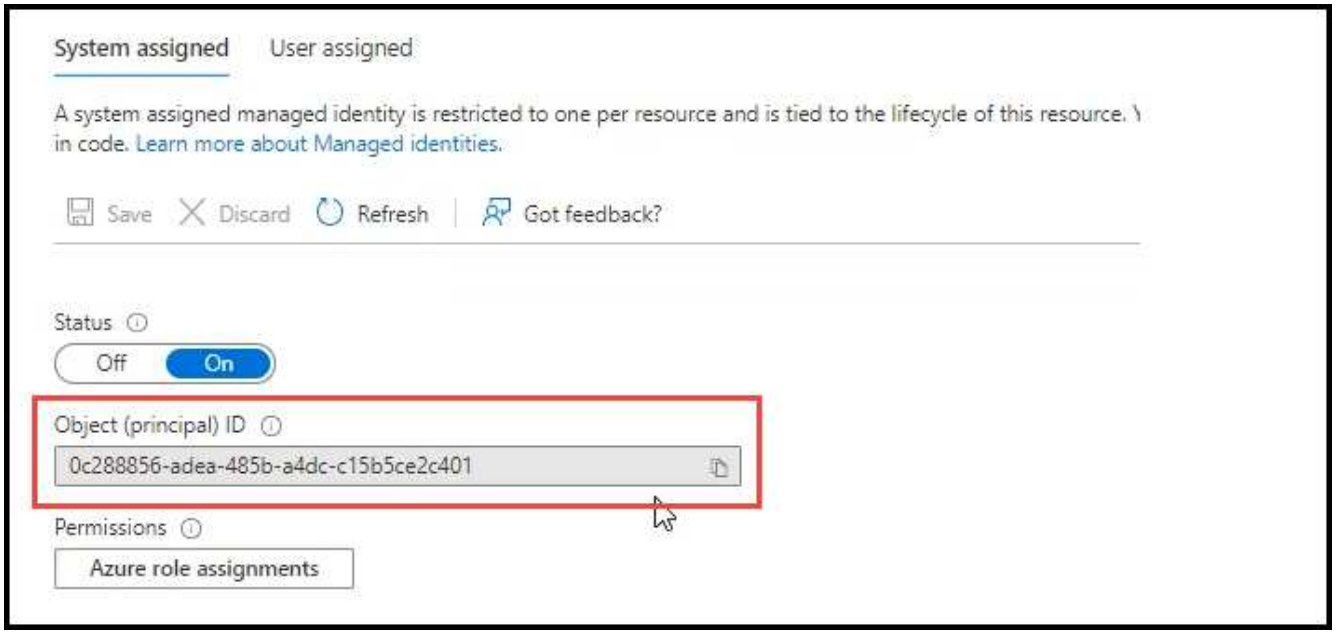
RBAC validation occurs only on Kubernetes clusters with Active Directory (AD) enabled. Kubernetes clusters without AD will pass validation automatically.

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

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.



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

Steps

1. Create a cluster role and role binding.
   a. Create a YAML file that includes the following text. 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.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
    name: cloudmanager-access-clusterrole
rules:
    - apiGroups:
        - ''
      resources:
        - secrets
        - namespaces
        - persistentvolumeclaims
        - persistentvolumes
      verbs:
        - get
        - list
        - create
    - apiGroups:
        - storage.k8s.io
      resources:
        - storageclasses
      verbs:
        - get
        - list
    - apiGroups:
        - trident.netapp.io
      resources:
        - tridentbackends
        - tridentorchestrators
      verbs:
        - get
        - list
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
    name: k8s-access-binding
subjects:
    - kind: User
      name: Object (principal) ID (for AKS) or username (for self-
managed)
      apiGroup: rbac.authorization.k8s.io
roleRef:
    kind: ClusterRole
    name: cloudmanager-access-clusterrole
    apiGroup: rbac.authorization.k8s.io
```

b. Apply the configuration to a cluster.

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

Add an Azure Kubernetes cluster to Cloud Manager

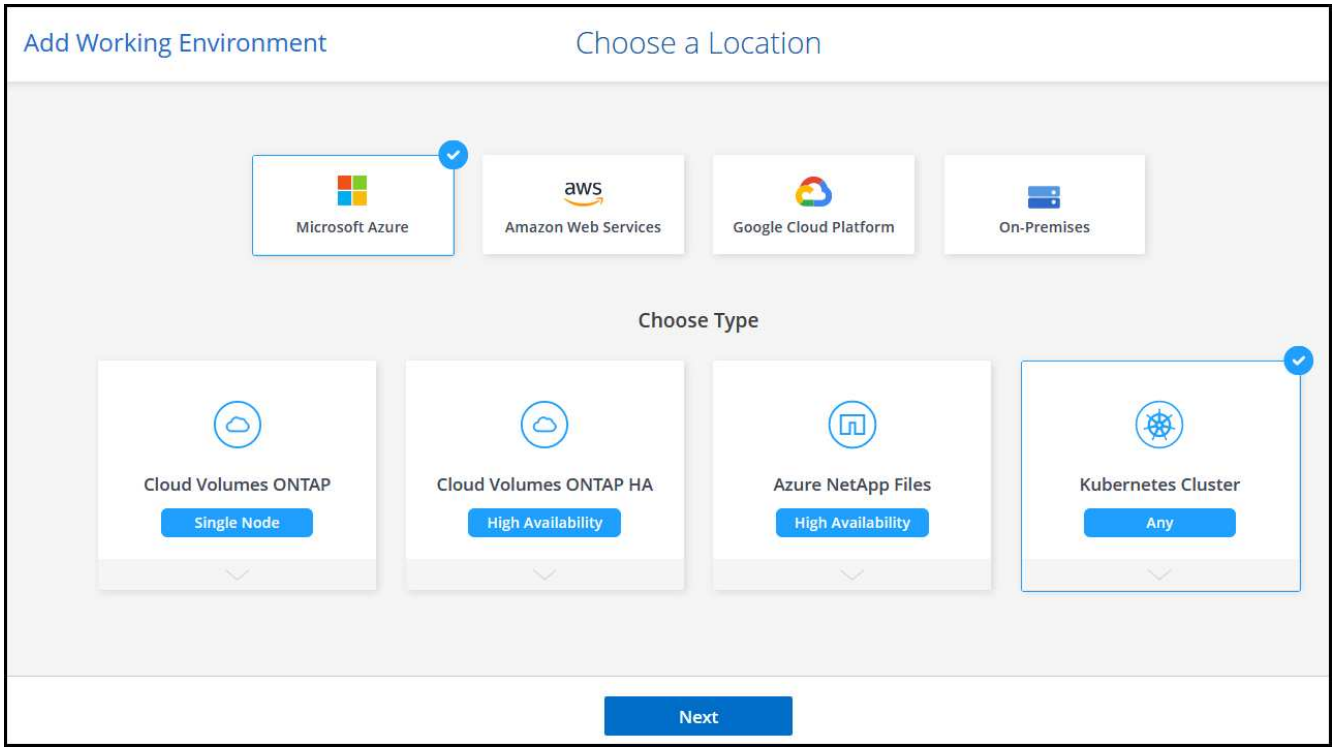
You can discover or import Kubernetes clusters to Cloud Manager so that you can back up persistent volumes to Azure.

Discover a cluster

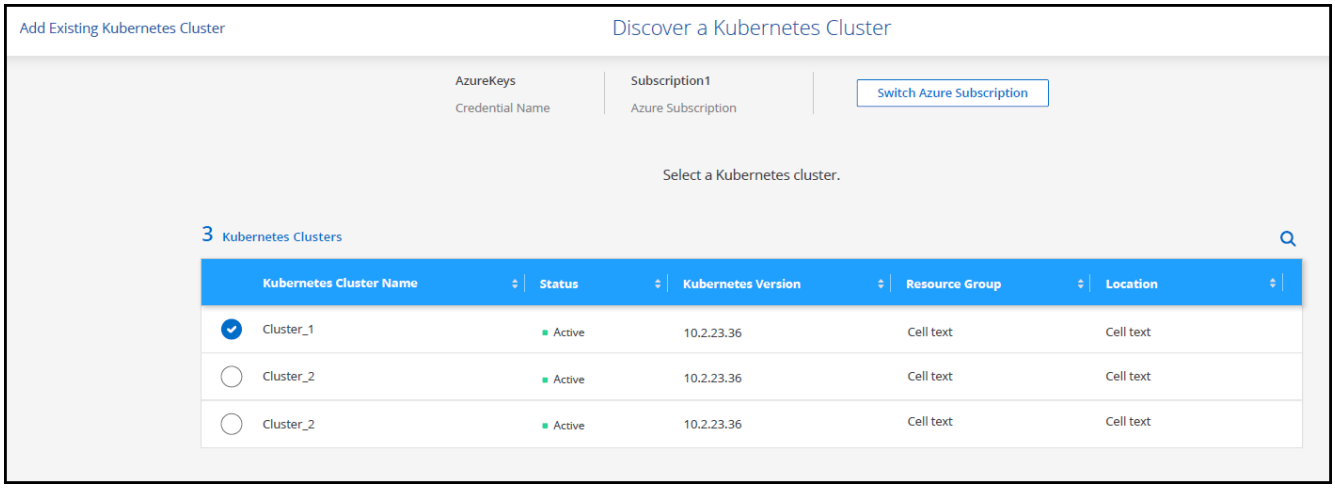
You can discover a fully-managed or self-managed Kubernetes cluster. Managed clusters must be discovered; they cannot be imported.

Steps

1. On the **Canvas**, click **Add Working Environment**.
2. Select **Microsoft Azure** > **Kubernetes Cluster** and click **Next**.

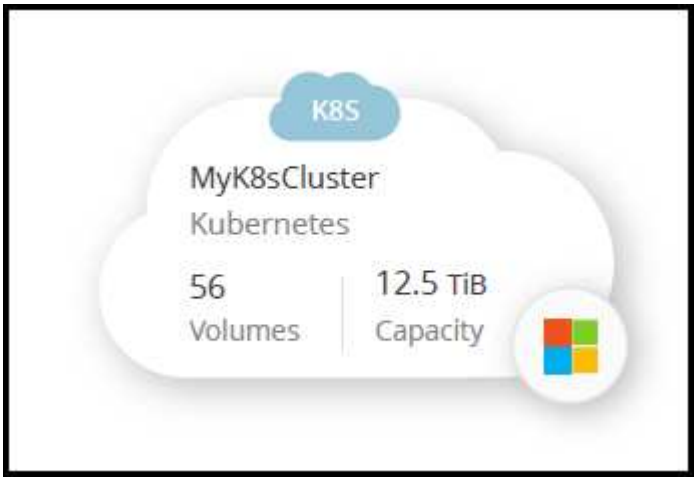


3. Select **Discover Cluster** and click **Next**.
4. Select a Kubernetes cluster and click **Next**.



Result

Cloud Manager adds the Kubernetes cluster to the Canvas.



Import a Cluster

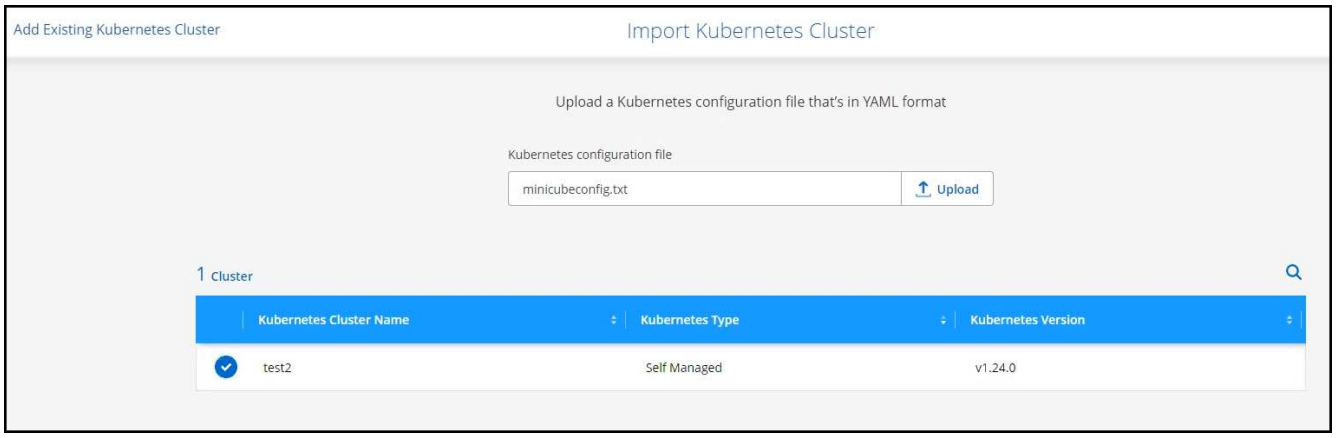
You can import a self-managed Kubernetes cluster using a Kubernetes configuration file.

Before you get started

You will need Certificate Authority, Client Key, and Client Certificate certificates for the user specified in the cluster role YAML file to import Kubernetes clusters. The Kubernetes cluster administrator receives these certifications when creating users on the Kubernetes cluster.

Steps

1. On the **Canvas**, click **Add Working Environment**.
2. Select **Microsoft Azure** > **Kubernetes Cluster** and click **Next**.
3. Select **Import Cluster** and click **Next**.
4. Upload a Kubernetes configuration file in YAML format.



5. Upload the cluster certificates provided by your Kubernetes cluster administrator.

Result

Cloud Manager adds the Kubernetes cluster to the Canvas.