Get started with Kubernetes clusters in AWS

Cloud Manager

NetApp
March 07, 2022

Table of Contents

Get started with Kubernetes clusters in AWS

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

The Kubernetes cluster must have NetApp Astra Trident installed. One of the four most recent versions of Astra Trident is required. Go to the Astra Trident docs for installation steps.

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.

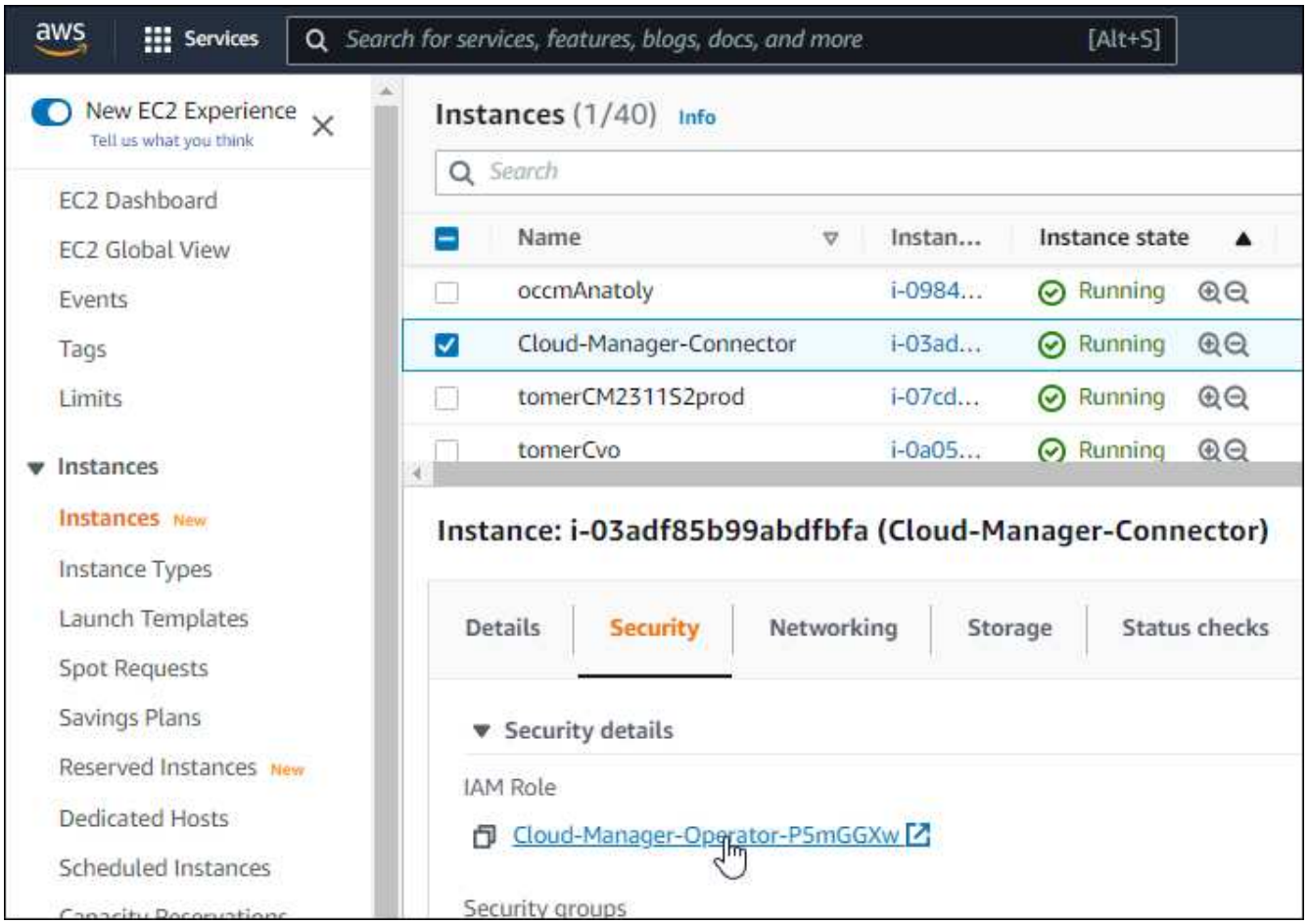
- Create a Connector from Cloud Manager (recommended)
- Create a Connector from the AWS Marketplace
- Install the Connector on an existing Linux host in AWS

Add the required permissions to an existing Connector

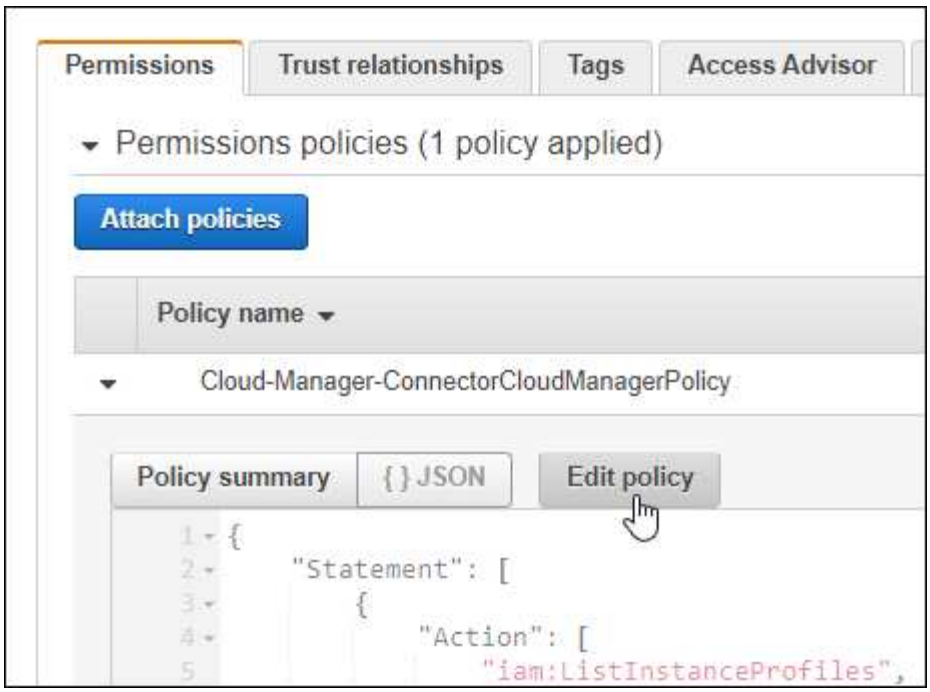
Starting in the 3.9.13 release, any *newly* created Connectors include three new AWS permissions that enable discovery and management of Kubernetes clusters. If you created a Connector prior to this release, then you'll need to modify the existing policy for the Connector's IAM role to provide the permissions.

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**.

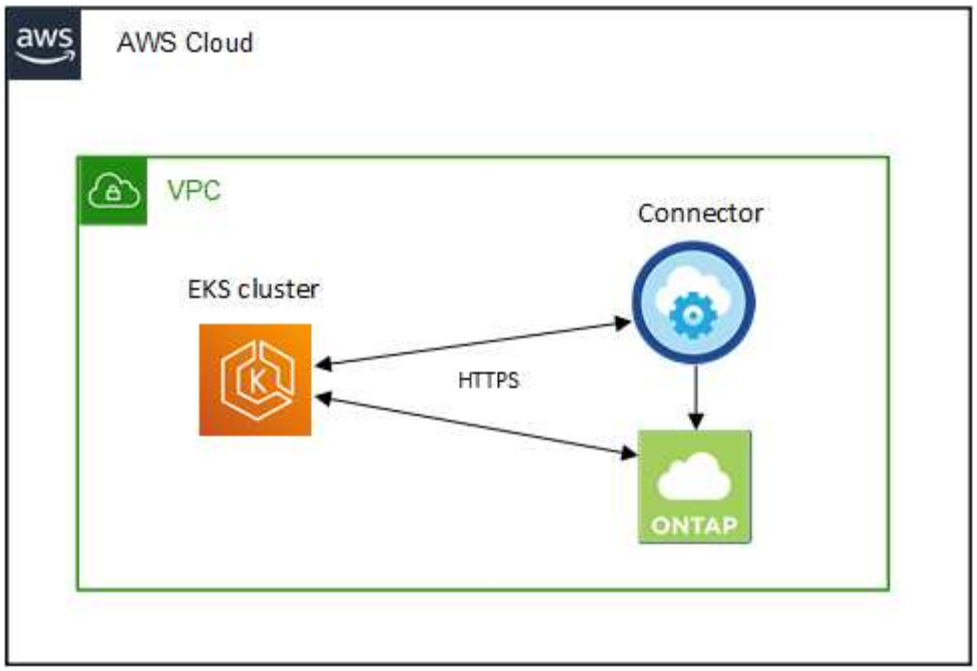
Review networking requirements

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.

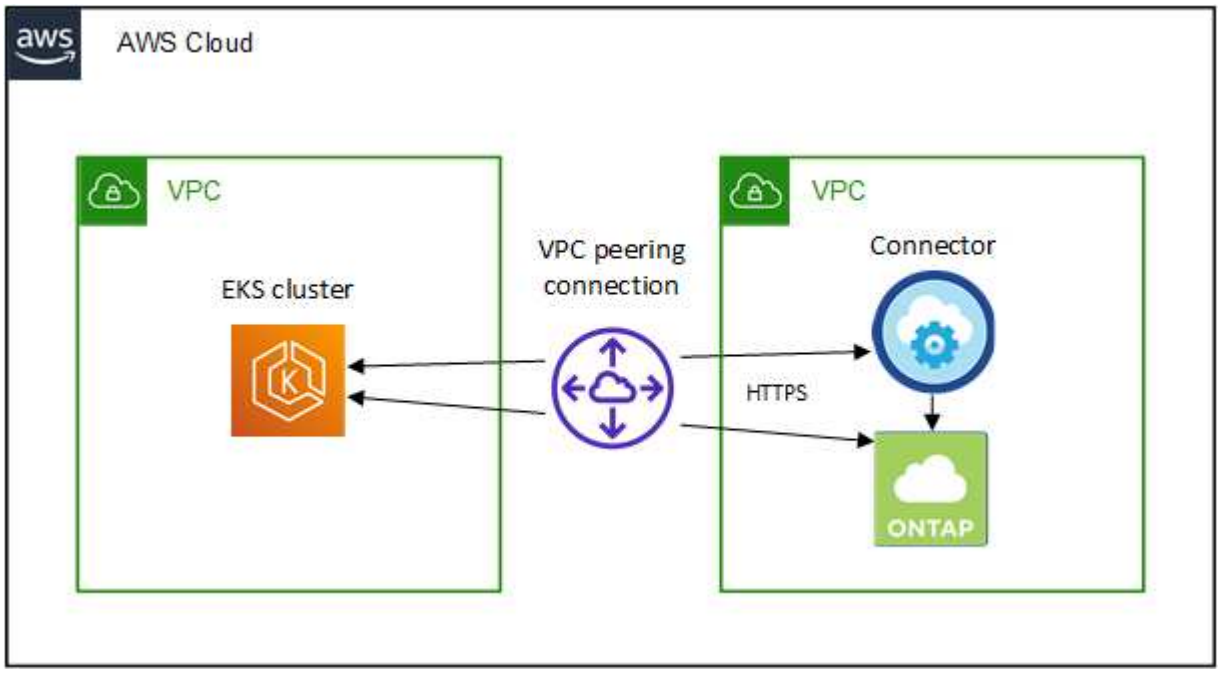
- Each Kubernetes cluster must have an inbound connection from the Connector
- The Connector must have an outbound connection to each Kubernetes cluster over port 443

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.

Steps

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

```
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: Group
      name: cloudmanager-access-group
      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>
```

2. Create an identity mapping to the permissions group.

Use eksctl

Use eksctl to create an IAM identity mapping between a cluster and the IAM role for the Cloud Manager Connector.

Go to the eksctl documentation for full instructions.

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}}
```

Edit aws-auth

Directly edit the aws-auth ConfigMap to add RBAC access to the IAM role for the Cloud Manager Connector.

Go to the AWS EKS documentation for full instructions.

An example is provided below.

```
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
```

Add an Amazon Kubernetes cluster to Cloud Manager

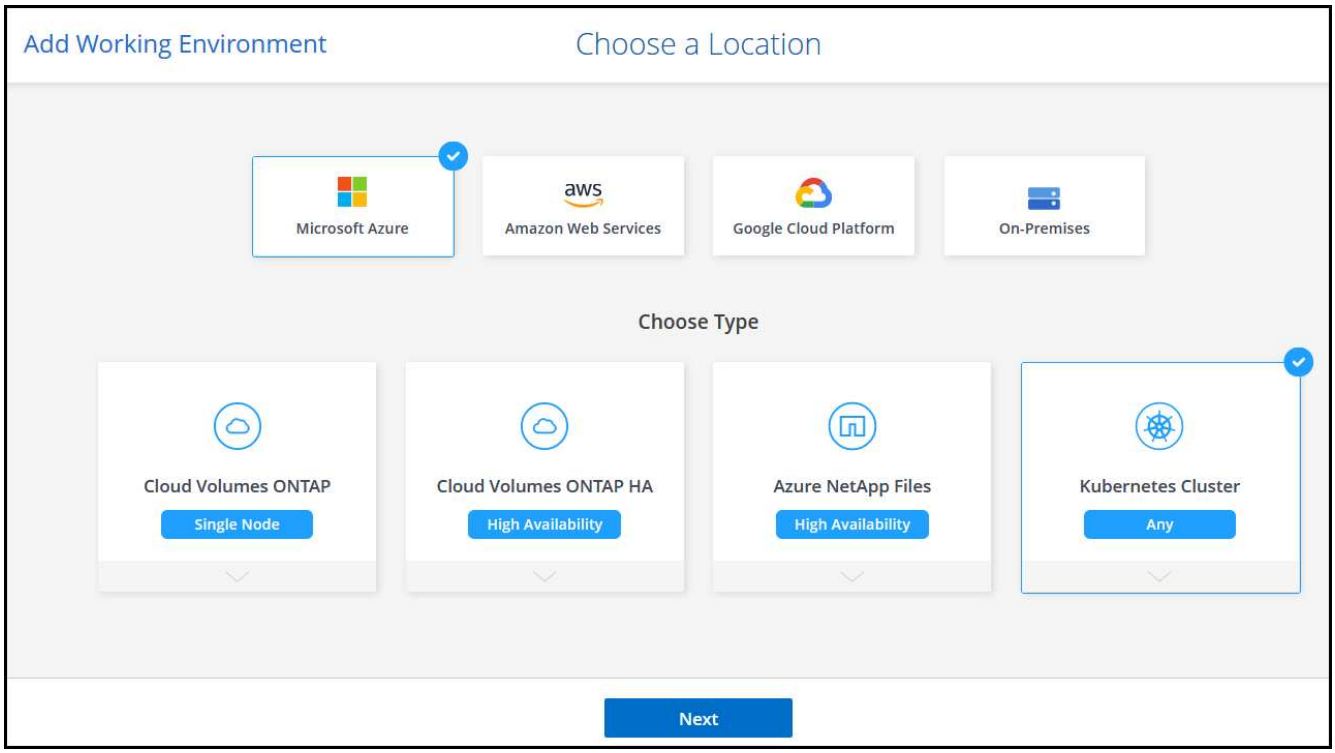
You can discover or import Kubernetes clusers to Cloud Manager so you can back up persistent volumes to Amazon S3.

Discover a cluster

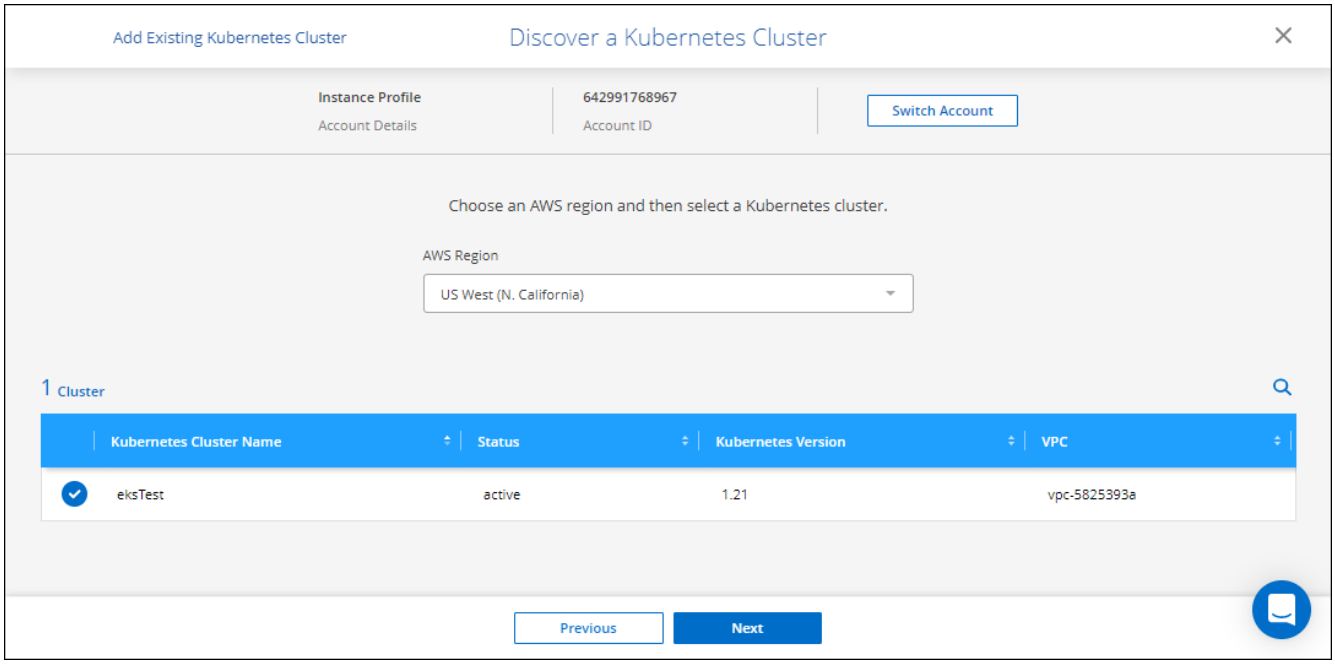
You can discover a fully-managed or self-managed Kubernetes cluster. Managed clusters must be discovered; they cannot be imported.

Steps

1. On the **Canvas**, click **Add Working Environment**.
2. Select **Amazon Web Services** > **Kubernetes Cluster** and click **Next**.

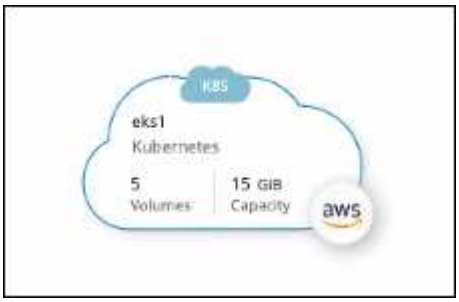


3. Select **Discover Cluster** and click **Next**.
4. Choose an AWS region, select a Kubernetes cluster, and then click **Next**.



Result

Cloud Manager adds the Kubernetes cluster to the Canvas.

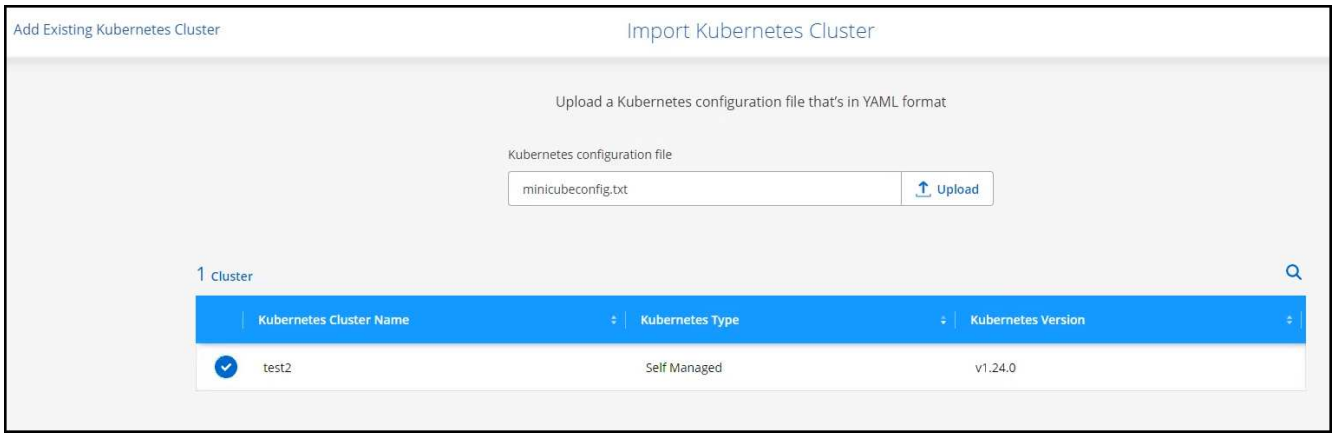


Import a Cluster

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

Steps

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



5. Select the Kubernetes cluster and click **Next**.

Result

Cloud Manager adds the Kubernetes cluster to the Canvas.