



Administering Clusters on Azure Kubernetes Service (AKS)

Integrating Cisco Container Platform with Azure Kubernetes Service (AKS) allows you to deploy and run containerized applications on the Azure cloud.

This chapter contains the following topics:

- [Prerequisites for Configuring AKS Clusters, on page 1](#)
- [Configuring Control Plane Proxy for AKS Access, on page 2](#)
- [Creating AKS Clusters, on page 2](#)
- [Upgrading AKS Clusters, on page 4](#)
- [Deleting AKS Clusters, on page 4](#)

Prerequisites for Configuring AKS Clusters

The prerequisites for configuring AKS clusters are as follows:

Azure User Account Requirements

The following roles are necessary for your Azure user account that you want to use with Cisco Container Platform:

- An **App administrator** within the directory
- An **Owner** within the Azure subscription

You can contact your Azure administrator to set up these roles for your Azure account.

Creating Resource Groups

A resource group is a logical grouping of the resources that are required to deploy Cisco Container Platform on Azure. They allow you to manage your resources efficiently.

Step 1 Log in to the [Microsoft Azure portal](#).

Step 2 In the left navigation pane, click **All services > Resource groups**.

The **Resource groups** pane is displayed.

Step 3 Click **Add** to create a new resource group.

Step 4 In the **Create an empty resource group** pane, specify the following information:

- In the **Resource Group name** field, enter a unique name.
- From the **Subscription** drop-down list, choose a subscription type.
- From the **Resource group location** drop-down list, choose the region in which you want to create your AKS cluster.

Note Not all regions support AKS. Ensure that you select a supported region. Currently, Cisco Container Platform supports only the **eastus** and **westus** regions.

For more information on a resource group, see [Resource Group Overview](#).

Configuring Control Plane Proxy for AKS Access

If your Control Plane VMs need proxy configuration to access the internet, specifically Azure API endpoints, you must configure the proxy information on Cisco Container Platform.

Step 1 SSH to the master node of the control plane.

Step 2 Run the following commands to specify the proxy information:

Note You need to replace `<Proxy_URL_or_IPAddress:Port>` with the URL/IP address of your proxy server and the `no_proxy` list with a list of your internal IP addresses.

```
kubectl patch deploy kaas-api
```

```
kubectl patch deploy kaas-ccp-aks-operator --patch
```

Creating AKS Clusters

Before you begin

- Ensure that you have added your Azure provider profile.
For more information, see [Adding Azure Provider Profile](#).
- Ensure that your Azure account that is used with Cisco Container Platform has the necessary roles.
For more information, see [Azure User Account Requirements, on page 1](#).
- Ensure that you have created the necessary resource groups in your Azure account.
For more information, see [Creating Resource Groups, on page 1](#).

Step 1 In the left pane, click **Clusters**, and then click the **Azure** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen, specify the following information:

- a) From the **INFRASTRUCTURE PROVIDER** drop-down list, choose the provider related to the appropriate Azure account.
- b) From the **AZURE REGION** drop-down list, choose an appropriate AKS region.
Note Not all regions support AKS. Ensure that you select a supported region. Currently, Cisco Container Platform supports only the **eastus** and **westus** regions.
- c) In the **KUBERNETES CLUSTER NAME** field, enter a name for your cluster.
- d) From the **KUBERNETES VERSION** drop-down list, choose the Kubernetes version for your cluster.
Note Not all Kubernetes versions are supported in all Azure regions. Ensure that you select a supported Kubernetes version.
- e) Click **NEXT**.

Step 4 In the **Cluster settings** screen, specify the following information:

- a) In the **NODE POOL NAME** field, enter an alphanumeric name for the primary node pool of your cluster.
- b) From the **RESOURCE GROUP** drop-down list, enter the name of the resource group that you have created in your Azure account.
See also [Creating Resource Groups, on page 1](#).
- c) To create and use a virtual network subnet for an AKS cluster, from the **NETWORK PLUGIN** drop-down list, choose **Azure** or **Kubenet**.
See also [Using Kubenet Networking](#).
- d) If you want to use Virtual Kubelet to provision pods on the Azure Container Instance in your AKS clusters, under **VIRTUAL KUBELET**, click the toggle button to enable virtual kubelet.
Note The provider must have an application name to enable Virtual Kubelet.
- e) Click **NEXT**.

Step 5 In the **Node Configuration** screen, specify the following information:

- a) From the **WORKER INSTANCE TYPE** drop-down list, choose an instance type that provides an appropriate combination of CPU, memory, storage, and networking capacity for your cluster.
See also [Amazon EC2 Instance Types](#).
- b) In the **WORKER COUNT** field, enter an appropriate number of worker nodes.
- c) In the **POD CIDR** field, enter the desired CIDR value.
Note This is an optional field. So you can leave the field empty or leave the default as is.
- d) In the **SERVICE CIDR** field, enter the desired CIDR value.
Note This is an optional field. So you can leave the field empty or leave the default as is.
- e) In the **VNET SUBNET ID** field, enter a subnet within the virtual network where you want to deploy the Azure resources.
- f) In the **DOCKER BRIDGE CIDR** field, enter a valid CIDR to override the default Docker bridge.
- g) In the **DNS SERVICE IP** field, enter an IP address within the Kubernetes service address range that will be used by cluster service discovery (kube-dns).
See also [Configure Azure CNI Networking in AKS](#).

Note You must not use the first IP address in your address range, such as .1. The first address in your subnet range is used for the `kubernetes.default.svc.cluster.local` address.

h) Click **NEXT**.

Step 6 In the **Summary** screen, review the cluster information, and then click **FINISH**.

Cluster creation can take up to 20 minutes. You can monitor cluster creation status on the **Clusters** screen.

Upgrading AKS Clusters

The upgrade path for an AKS cluster is allowed based on the AKS upgrade map, which is completely controlled by AKS.

For example:

If the current version of your AKS cluster is 1.12.x, you must first upgrade 1.12.x to 1.13.x, and then upgrade from 1.13.x to 1.14.x. You cannot directly upgrade from 1.12.x to 1.14.x.

See also [Upgrade an AKS cluster](#).

Before you begin

Ensure that the AKS cluster is in the **READY** state.

-
- Step 1** In the left pane, click **Clusters**, and then click the **Azure** tab.
 - Step 2** From the drop-down list displayed under the **ACTIONS** column, choose **Upgrade** for the cluster that you want to upgrade.
 - Step 3** In the **Upgrade Azure Cluster** dialog box, choose a Kubernetes version for the VM, and then click **Upgrade**. The cluster status changes from **READY** to **UPGRADING** to **READY**.
It may take a few minutes for the Kubernetes cluster upgrade to complete.
-

Deleting AKS Clusters

Before you begin

Ensure that the AKS cluster that you want to delete is not currently in use, as deleting a cluster removes the containers and data associated with it.

-
- Step 1** In the left pane, click **Clusters**, and then click the **Azure** tab.
 - Step 2** From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.
 - Step 3** Click **DELETE** in the confirmation dialog box.
-

Upon deleting an AKS cluster, it takes about 15 minutes for the cluster resources to be released.