Azure-Kubernetes-Service

Core concepts for Azure Kubernetes Service (AKS)

This article describes core concepts of Azure Kubernetes Service (AKS), a managed Kubernetes service that you can use to deploy and operate containerized applications at scale on Azure.

What is Kubernetes?

Kubernetes is an open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications. For more information, see the official <u>Kubernetes documentation</u>.

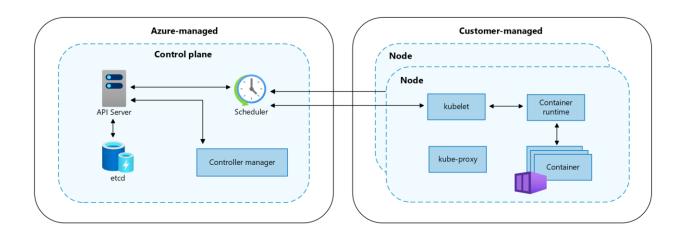
What is AKS?

AKS is a managed Kubernetes service that simplifies deploying, managing, and scaling containerized applications using Kubernetes. For more information, see What is Azure Kubernetes Service (AKS)?

Cluster components

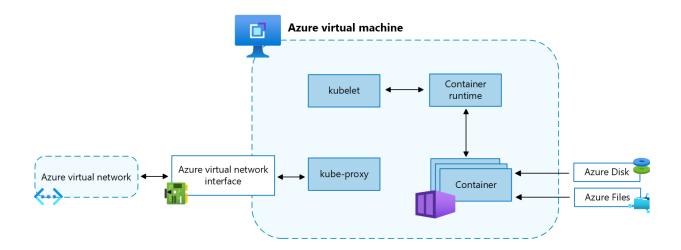
An AKS cluster is divided into two main components:

- Control plane: The control plane provides the core Kubernetes services and orchestration of application workloads.
- Nodes: Nodes are the underlying virtual machines (VMs) that run your applications.



Nodes

Each AKS cluster has at least one node, which is an Azure virtual machine (VM) that runs Kubernetes node components. The following components run on each node:



Node configuration

VM size and image

The **Azure VM size** for your nodes defines CPUs, memory, size, and the storage type available, such as high-performance SSD or regular HDD. The VM size you choose depends on the workload requirements and the number of pods you plan to run on each node. For more information, see <u>Supported VM sizes in Azure Kubernetes Service (AKS)</u>.

In AKS, the **VM image** for your cluster's nodes is based on Ubuntu Linux, <u>Azure Linux</u>, or Windows Server 2022. When you create an AKS cluster or scale out the number of nodes, the Azure platform automatically creates and configures the requested number of VMs. Agent nodes are billed as standard VMs, so any VM size discounts, including <u>Azure reservations</u>, are automatically applied.

OS disks

Default OS disk sizing is only used on new clusters or node pools when Ephemeral OS disks aren't supported and a default OS disk size isn't specified. For more information, see <u>Default OS disk sizing</u> and <u>Ephemeral OS disks</u>.

Resource reservations

AKS uses node resources to help the nodes function as part of the cluster. This usage can cause a discrepancy between the node's total resources and the allocatable resources in AKS. To maintain node performance and functionality, AKS reserves two types of resources, **CPU** and **memory**, on each node. For more information, see <u>Resource</u> reservations in AKS.

OS

AKS supports Ubuntu 22.04 and Azure Linux 2.0 as the node OS for Linux node pools. For Windows node pools, AKS supports Windows Server 2022 as the default OS. Windows Server 2019 is being retired after Kubernetes version 1.32 reaches end of life and isn't supported in future releases. If you need to upgrade your Windows OS version, see Upgrade from Windows Server 2019 to Windows Server 2022. For more information on using Windows Server on AKS, see Windows Container Considerations in Azure Kubernetes Service (AKS).

Container runtime

A container runtime is software that executes containers and manages container images on a node. The runtime helps abstract away sys-calls or OS-specific functionality to run containers on Linux or Windows. For Linux node pools, <u>containerd</u> is used on Kubernetes version 1.19 and higher. For Windows Server 2019 and 2022 node pools, <u>containerd</u> is generally available and is the only runtime option on Kubernetes version 1.23 and higher.

Pods

A *pod* is a group of one or more containers that share the same network and storage resources and a specification for how to run the containers. Pods typically have a 1:1 mapping with a container, but you can run multiple containers in a pod.

Node pools

In AKS, nodes of the same configuration are grouped together into *node pools*. These node pools contain the underlying virtual machine scale sets and virtual machines (VMs) that run your applications. When you create an AKS cluster, you define the initial number of nodes and their size (SKU), which creates a *system node pool*. System node pools serve the primary purpose of hosting critical system pods, such as CoreDNS and konnectivity. To support applications that have different compute or storage demands, you can create *user node pools*. User node pools serve the primary purpose of hosting your application pods.

For more information, see Create node pools in AKS and Manage node pools in AKS.

Node resource group

When you create an AKS cluster in an Azure resource group, the AKS resource provider automatically creates a second resource group called the *node resource group*. This resource group contains all the infrastructure resources associated with the cluster, including virtual machines (VMs), virtual machine scale sets, and storage.

For more information, see the following resources:

- Why are two resource groups created with AKS?
- Can I provide my own name for the AKS node resource group?
- Can I modify tags and other properties of the resources in the AKS node resource group?

Namespaces

Kubernetes resources, such as pods and deployments, are logically grouped into *namespaces* to divide an AKS cluster and create, view, or manage access to resources.

The following namespaces are created by default in an AKS cluster:

Expand table

Namespace	Description
default	The <u>default</u> namespace allows you to start using cluster resources without creating a new namespace.
kube-node- lease	The <u>kube-node-lease</u> namespace enables nodes to communicate their availability to the control plane.
kube-public	The <u>kube-public</u> namespace isn't typically used, but can be used for resources to be visible across the whole cluster by any user.
kube-system	The <u>kube-system</u> namespace is used by Kubernetes to manage cluster resources, such as coredns, konnectivity-agent, and metrics-server.

