Azure Kubernetes Service (AKS)

• 5 minutes to read • +22

Azure Kubernetes Service (AKS) simplifies deploying a managed Kubernetes cluster in Azure by offloading much of the complexity and operational overhead to Azure. As a hosted Kubernetes service, Azure handles critical tasks for you, like health monitoring and maintenance.

Since the Kubernetes masters are managed by Azure, you only manage and maintain the agent nodes. Thus, as a managed Kubernetes service, AKS is free; you only pay for the agent nodes within your clusters, not for the masters.

You can create an AKS cluster using the Azure portal, the Azure CLI, Azure PowerShell, or using template-driven deployment options, such as Resource Manager templates and Terraform. When you deploy an AKS cluster, the Kubernetes master and all nodes are deployed and configured for you. Additional features such as advanced networking, Azure Active Directory integration, and monitoring can also be configured during the deployment process. Windows Server containers are supported in AKS.

For more information on Kubernetes basics, see Kubernetes core concepts for AKS.

To get started, complete the AKS Quickstart in the Azure portal or with the Azure CLI.

Note

This service supports Azure Lighthouse, which lets service providers sign in to their own tenant to manage subscriptions and resource groups that customers have delegated.

Access, security, and monitoring

For improved security and management, AKS lets you integrate with Azure Active Directory (Azure AD) and:

- Use Kubernetes role-based access control (Kubernetes RBAC).
- Monitor the health of your cluster and resources.

Identity and security management

To limit access to cluster resources, AKS supports Kubernetes RBAC. Kubernetes RBAC lets you control access and permissions to Kubernetes resources and namespaces.

You can also configure an AKS cluster to integrate with Azure AD. With Azure AD integration, you can configure Kubernetes access based on existing identity and group membership. Your existing Azure AD users and groups can be provided with an integrated sign-on experience and access to AKS resources.

For more information on identity, see Access and identity options for AKS.

To secure your AKS clusters, see Integrate Azure Active Directory with AKS.

Integrated logging and monitoring

Azure Monitor for Container Health collects memory and processor performance metrics from containers, nodes, and controllers within your AKS cluster and deployed applications. You can review both the container logs and the Kubernetes master logs. This monitoring data is stored in an Azure Log Analytics workspace and is available through the Azure portal, Azure CLI, or a REST endpoint.

For more information, see Monitor Azure Kubernetes Service container health.

Clusters and nodes

AKS nodes run on Azure virtual machines (VMs). With AKS nodes, you can connect storage to nodes and pods, upgrade cluster components, and use GPUs. AKS supports Kubernetes clusters that run multiple node pools to support mixed operating systems and Windows Server containers.

For more information regarding Kubernetes cluster, node, and node pool capabilities, see Kubernetes core concepts for AKS.

Cluster node and pod scaling

As demand for resources change, the number of cluster nodes or pods that run your services can automatically scale up or down. You can use both the horizontal pod autoscaler or the cluster autoscaler. This approach to scaling lets the AKS cluster automatically adjust to demands and only run the resources needed.

For more information, see Scale an Azure Kubernetes Service (AKS) cluster.

Cluster node upgrades

AKS offers multiple Kubernetes versions. As new versions become available in AKS, your cluster can be upgraded using the Azure portal or Azure CLI. During the upgrade process, nodes are carefully cordoned and drained to minimize disruption to running applications.

To learn more about lifecycle versions, see Supported Kubernetes versions in AKS. For steps on how to upgrade, see Upgrade an Azure Kubernetes Service (AKS) cluster.

GPU-enabled nodes

AKS supports the creation of GPU-enabled node pools. Azure currently provides single or multiple GPU-enabled VMs. GPU-enabled VMs are designed for compute-intensive, graphics-intensive, and visualization workloads.

For more information, see Using GPUs on AKS.

Confidential computing nodes (public preview)

AKS supports the creation of Intel SGX-based, confidential computing node pools (DCSv2 VMs). Confidential computing nodes allow containers to run in a hardware-based, trusted execution environment (enclaves). Isolation between containers, combined with code integrity through attestation, can help with your defense-in-depth container security strategy. Confidential computing nodes support both confidential containers (existing Docker apps) and enclave-aware containers.

For more information, see Confidential computing nodes on AKS.

Storage volume support

To support application workloads, you can mount storage volumes for persistent data. You can use both static and dynamic volumes. Depending on the number of connected pods expected to share the storage volumes, you can use storage backed by either Azure Disks for single pod access, or Azure Files for multiple concurrent pod access.

For more information, see Storage options for applications in AKS.

Get started with dynamic persistent volumes using Azure Disks or Azure Files.

Virtual networks and ingress

An AKS cluster can be deployed into an existing virtual network. In this configuration, every pod in the cluster is assigned an IP address in the virtual network, and can directly communicate with other pods in the cluster and other nodes in the virtual network. Pods can also connect to other services in a peered virtual network and to on-premises networks over ExpressRoute or site-to-site (S2S) VPN connections.

For more information, see the Network concepts for applications in AKS.

Ingress with HTTP application routing

The HTTP application routing add-on makes it easy to access applications deployed to your AKS cluster. When enabled, the HTTP application routing solution configures an ingress controller in your AKS cluster.

As applications are deployed, publicly accessible DNS names are autoconfigured. The HTTP application routing sets up a DNS zone and integrates it with the AKS cluster. You can then deploy Kubernetes ingress resources as normal.

To get started with ingress traffic, see HTTP application routing.

Development tooling integration

Kubernetes has a rich ecosystem of development and management tools that work seamlessly with AKS. These tools include Helm and the Kubernetes extension for Visual Studio Code. These tools work seamlessly with AKS.

Additionally, Azure provides several tools that help streamline Kubernetes, such as DevOps Starter.

DevOps Starter provides a simple solution for bringing existing code and Git repositories into Azure. DevOps Starter automatically:

- Creates Azure resources (such as AKS);
- Configures a release pipeline in Azure DevOps Services that includes a build pipeline for CI;

- Sets up a release pipeline for CD; and,
- Generates an Azure Application Insights resource for monitoring.

For more information, see DevOps Starter.

Docker image support and private container registry

AKS supports the Docker image format. For private storage of your Docker images, you can integrate AKS with Azure Container Registry (ACR).

To create a private image store, see Azure Container Registry.

Kubernetes certification

AKS has been CNCF-certified as Kubernetes conformant.

Regulatory compliance

AKS is compliant with SOC, ISO, PCI DSS, and HIPAA. For more information, see Overview of Microsoft Azure compliance .

Next steps

Learn more about deploying and managing AKS with the Azure CLI Quickstart.

AKS quickstart

Is this page helpful?

Yes No