



# How Azure Kubernetes Service works

5 minutes

Now that you know the basics of Azure Kubernetes Service, let's see what information you need to set up a simple AKS cluster. This information should help you decide how much additional configuration AKS may require when integrating with existing development and deployment processes.

## Creating an AKS cluster

At its core, an AKS cluster is a cloud hosted Kubernetes cluster. Unlike a custom Kubernetes installation, AKS streamlines the installation process and takes care of most of the underlying cluster management tasks.

You have two options when you create an AKS cluster. You either use the Azure portal or Azure CLI. Both options require you to configure basic information about the cluster. For example:

- The Kubernetes cluster name
- The version of Kubernetes to install
- A DNS prefix to make the master node publicly accessible
- The initial node pool size

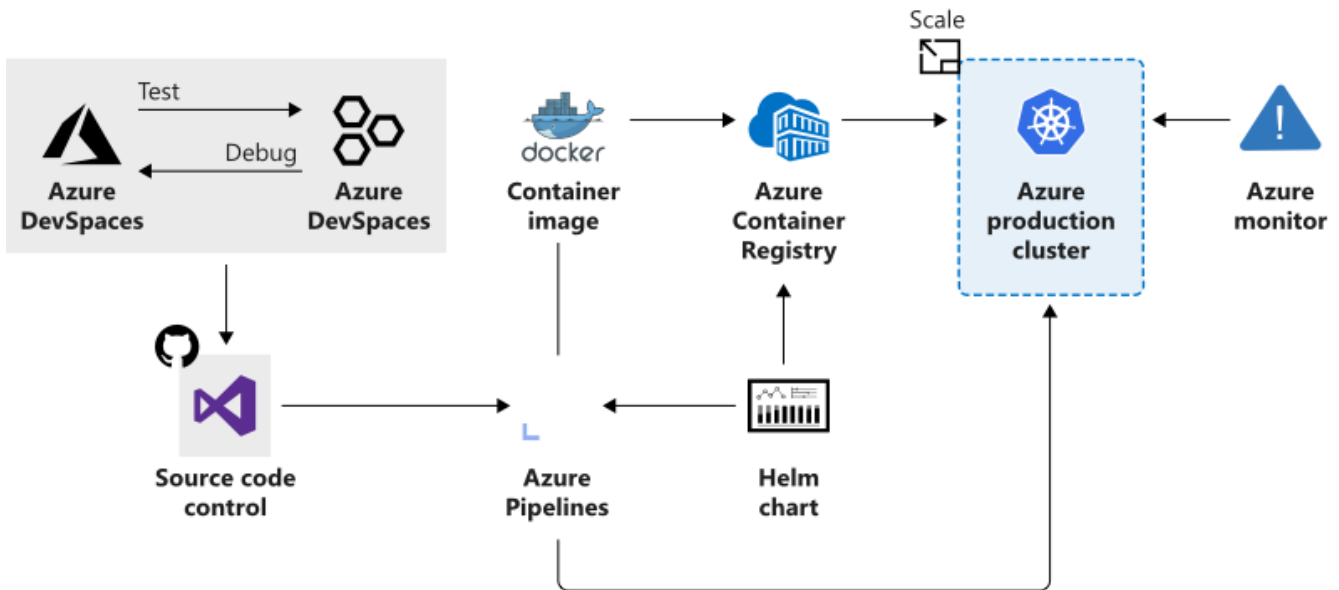
The initial node pool size defaults to two nodes, however it's recommended that at least three nodes are used for a production environment.

### Note

The master node in your cluster is free. You only pay for node VMs, storage and networking resources consumed in your cluster.

Unless specified, the Azure service creation workflow creates a Kubernetes cluster using default configuration for scaling, authentication, networking and monitoring. Creating an AKS cluster typically takes a few minutes. Once complete, you can change any of the default AKS cluster properties. Access and management of your cluster is done through the Azure portal or from the command line.

# How workloads are developed and deployed to AKS



AKS supports the Docker image format that means that you can use any development environment to create a workload, package the workload as a container and deploy the container as a Kubernetes pod.

Here you use the standard Kubernetes command-line tools or the Azure CLI to manage your deployments. The support for the standard Kubernetes tools ensures that you don't need to change your current workflow to support an existing Kubernetes migration to AKS.

AKS also supports all the popular development and management tools such as Helm, Draft, Kubernetes extension for Visual Studio Code and Visual Studio Kubernetes Tools.

## Azure Dev Spaces

Setting up a local Kubernetes cluster on a developer machine can be complex and most solutions offers a single node configuration. It's also common to mock or replicate dependencies between developer teams when working on microservices projects.

Azure Dev Spaces helps your development teams be more productive on Kubernetes and allows you to:

- Minimize the local dev machine setup for each team member as developers can work directly in AKS
- Rapidly iterate and debug code directly in Kubernetes using Visual Studio or Visual Studio Code
- Generate Docker and Kubernetes configuration-as-code assets to use from development through to production
- Develop your code in isolation, and do integrated testing with other components without replicating or mocking up dependencies

**i Important**

Azure Dev Spaces is supported only by AKS clusters in specific regions.

## Deployment Center

Deployment center simplifies setting up a DevOps pipeline for your application. You can use this configured DevOps pipeline to set up a continuous integration (CI) and continuous delivery (CD) pipeline to your AKS Kubernetes cluster.

With Azure DevOps Projects you can:

- Automatically create Azure resources, such as an AKS cluster
- Create an Azure Application Insights resource for monitoring an AKS cluster
- Enable Azure Monitor for containers to monitor performance for the container workloads on an AKS cluster

You can add richer DevOps capabilities by extending the default configured DevOps pipeline. For example, you can add approvals before deploying, provision additional Azure resources, run scripts or upgrade workloads.

## Azure Service Integration

AKS allows us to integrate any Azure service offering and use it as part of an AKS cluster solution.

For example, remember that Kubernetes doesn't provide middleware and storage systems. Suppose you need to add a processing queue to the fleet management data processing service. You can easily integrate Storage queues using Azure Storage to extend the capacity of the data processing service.

---

**Next unit: When to use Azure Kubernetes Service**

Continue >