

CLOUD, DEVOPS, AND CONTAINERS

Deploying to Azure Kubernetes Service (AKS)

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

Kubernetes is powerful and famously demanding to operate, which is exactly why managed offerings like Azure Kubernetes Service exist: Azure runs the control plane so your team can concentrate on workloads. But managed does not mean hands-off. AKS still asks you to make real decisions about node pools, networking, identity, scaling, and upgrades, and the teams that struggle with it are usually the ones who never learned which decisions matter.

This is a hands-on, practitioner course. Rather than attempt all of Kubernetes and all of Azure at once, it goes deep on the path most teams actually walk: standing up a well-configured AKS cluster, deploying real applications to it, exposing them safely, and then scaling and operating them in production. Every module includes a lab against a live cluster, and each module builds on the one before.

Who Should Attend

- Engineers and DevOps practitioners deploying containerized applications on Azure
- Teams moving workloads from VMs or App Service to Kubernetes
- Kubernetes users from other platforms adopting AKS

Learners new to Kubernetes itself should take *Kubernetes Fundamentals* first.

Prerequisites

- Working knowledge of containers and images; *Introduction to Docker and Containers* covers it
- Kubernetes basics (pods, deployments, services), per *Kubernetes Fundamentals*
- Basic Azure familiarity and comfort at a command line

What You Will Learn

- Design and create an AKS cluster with sensible choices for node pools, networking, and identity
- Integrate AKS with Azure Container Registry and deploy applications to the cluster
- Expose services safely with ingress and TLS
- Configure scaling at every level: pods, nodes, and event-driven workloads
- Package and release applications with Helm
- Operate a cluster: upgrades, monitoring, access control, and cost management

Course Outline

Day one: standing up the cluster and shipping to it

- Designing the Cluster
 - What AKS manages for you, and what remains yours
 - Node pools, VM sizes, and availability zones
 - The networking choices you cannot easily undo

- Lab: create an AKS cluster with the CLI and inspect what Azure built
- Deploying Workloads
 - Connecting AKS to Azure Container Registry
 - Configuration and secrets on AKS, including Key Vault integration
 - Lab: deploy a containerized application from ACR to the cluster
- Exposing Services
 - Services, ingress controllers, and the Azure options
 - TLS and DNS for real endpoints
 - Lab: publish the application behind ingress with TLS

Day two: scaling and operating in production

- Scaling
 - Horizontal pod autoscaling and resource requests done right
 - The cluster autoscaler and node pool strategy
 - Event-driven scaling with KEDA, briefly
 - Lab: load-test the application and tune autoscaling until it holds
- Releasing with Helm
 - Charts, values, and releases: packaging an application properly
 - Upgrades and rollbacks you can trust
 - Lab: package the application as a Helm chart and roll a release back
- Operating the Cluster
 - Upgrades: cluster versions and node images without downtime
 - Monitoring with Azure Monitor and container insights
 - Access control with Entra ID and Kubernetes RBAC, and controlling cost
 - Lab: upgrade the cluster and diagnose a failing workload from monitoring data

Extended Version

The three-day version keeps the same gradient and adds:

- GitOps-style deployment to AKS and pipeline integration
- Network policy, private clusters, and a deeper security pass
- Multi-environment strategy: dev, staging, and production clusters
- A capstone: take an application from container image to a scaled, monitored, TLS-secured production deployment on AKS