

## Kubernetes with Azure & DevOps Workshop

2 Days / Instructor-Led / Format: On-Site or Virtual

Contact Us for Pricing

**About this Course:** Kubernetes has become the de facto standard for managing containerized applications by automating the processes for deploying, scaling, and managing containers.

Learn the fundamentals of Kubernetes and the corresponding Azure services for developers and administrators.

### Key Learning Areas:

- Understand the features, design goals and architecture of Kubernetes
- Install the tools necessary to work with Kubernetes
- Create and manage resources using Kubernetes commands
- Specify and create resources using yaml files
- Deploy applications to Kubernetes
- Scale, update and rollback Kubernetes deployments
- Create services to communicate with Kubernetes applications
- Expose services using Ingress
- Store and share images using the Azure Container Registry (ACR)
- Create a simple app deployment with an Azure Container Instance (ACI)
- Deploy a fully managed Kubernetes-style application with Azure Container Apps
- Leverage the full set of deployment features with Azure Kubernetes Service (AKS)
- Use Azure to automatically create a DevOps pipeline for Azure Container Apps and AKS

**Who Should Attend:** Developers and administrators who would like to leverage the advantages of containers by learning the fundamentals of Azure Kubernetes.

**Prerequisites:** Familiarity with Docker containers and basic Linux commands.

## Course Outline:

### Introduction to Kubernetes

For production scenarios, it is not enough to merely containerize an application. The containers need to be managed to provide features such as load balancing, scaling, service discovery, automated rollout and rollback of deployments, matching containers to node resources, and automatically replacing failed or unhealthy containers. This module introduces Kubernetes, which has become the standard platform for managing containers and providing the features required for real-world application deployment.

- What is Kubernetes?
- Architecture Overview
  - Cluster
  - Nodes
  - Pods
  - Control plane
- Tools, Installation, and Basic Commands
  - Minikube
  - Kubectl
- Creating a simple "Hello World" deployment

### Kubernetes Deployment

This module begins with detailed coverage of the architectural details of Kubernetes that are necessary for developing a thorough understand of Kubernetes deployments. It then demonstrates how to create a deployment from both the command line and using a yaml file. The module concludes with a discussion of how to update, rollback, scale, pause and resume a deployment.

- Architecture details
  - Pods
  - Nodes
  - Objects/Resources
  - Workloads
  - ReplicaSets
- Deployment
  - Create from the command line
  - Create from a yaml file

- Update
- Rollback
- Scaling
- Pause/resume

## Kubernetes Networking Part 1

Applications deployed into Kubernetes typically need to access and be accessible to other applications. This module presents the facilities provided by Kubernetes for communicating with a pod directly and by using a service. The details of services are presented in detail, with a discussion of how to select a service type, how to create services implicitly from the command line and explicitly from a yaml file, and how to access a service from within a cluster, from outside of a cluster, and from a remote machine.

- Networking model
- Services
  - Selectors and labels
  - Endpoints
  - Types
    - ClusterIP
    - NodePort
    - LoadBalancer
    - External name
  - Creation
    - CLI
    - Yaml
  - Exposing
    - Within the cluster
    - Externally
    - Local port forwarding

## Kubernetes Networking Part 2

Ingress represents the recommended way of externally exposing Kubernetes services. This module demonstrates how to use an Ingress resource to centralize routing rules so that traffic is sent to the appropriate Kubernetes services. Common Ingress usage patterns are presented, specifically fanout and name-based virtual hosting.

- What is Ingress?
- Creating an Ingress resource
  - Defining an Ingress resource in yaml
- Using an Ingress controller
- Ingress DNS
- Usage patterns
  - Fanout
  - Name-based virtual hosting

## Azure Container Registry/Instance (ACR/ACI)

Azure provides a wide range of resources for supporting containerized applications. This module shows how to store and share images in Azure using the Azure Container Registry (ACR) and how to run containers in the simplest and cheapest fashion using an Azure Container Instance (ACI).

- Azure Container Registry (ACR)
  - Introduction
  - Creation
  - Login
  - Pushing an image
  - Viewing an image
- Azure Container Instance (ACI)
  - Introduction
  - Creation/Configuration
  - Testing

## Azure Container Apps

Azure provides a container resource that occupies the middle ground in terms of complexity and expense. Azure Container Apps provide a way of running a containerized application that is more flexible than the ACI while being much simpler to configure than AKS.

- Azure Container Apps
  - Creation/Configuration
  - Container App Environment

- Ingress
- Testing

## **Azure Kubernetes Services (AKS)**

AKS represents the most fully featured way of running a container in Azure. It provides a myriad of features related to access, networking, security, health monitoring and Continuous Integration/Continuous Deployment (CI/CD). This module discusses the available options for creating an AKS cluster and how to use it for deploying, running, scaling and updating an application.

- What is AKS?
- Create an AKS Cluster
- Manage an application in AKS
  - Run
  - Scale
  - Update