

CLOUD, DEVOPS, AND CONTAINERS

Building Applications on Azure (App Service and Functions)

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

Overview

Most applications do not need a Kubernetes cluster. Azure App Service and Azure Functions can run a remarkable share of real-world workloads with far less operational burden, yet many teams either overlook them or use a fraction of what they offer. The hard part is knowing what the platform will do for you, and designing your application to let it.

This is a hands-on, practitioner course. Rather than survey every compute option Azure offers, it goes deep on the two managed services most application teams should reach for first, covering the core of the AZ-204 compute material along the way. The gradient runs from choosing the right compute for a workload, through deploying and configuring real applications on App Service, to building event-driven serverless components with Functions, and finally to securing, monitoring, and operating what you have built. Every module includes a lab, and each module builds on the one before.

Who Should Attend

- Developers building or migrating applications to Azure
- Teams choosing between App Service, Functions, containers, and Kubernetes
- Engineers preparing for AZ-204 who want depth on its core compute services

Prerequisites

- Working proficiency in at least one language Azure supports well (C#, JavaScript, Python, or Java)
- Basic Azure familiarity; true beginners should take *Microsoft Azure Fundamentals (AZ-900)* first
- Comfort with git and basic HTTP concepts

What You Will Learn

- Judge which Azure compute service fits a given workload, and when App Service or Functions is enough
- Deploy applications to App Service with deployment slots and safe, zero-downtime releases
- Configure applications the platform way: app settings, managed identity, and Key Vault
- Build event-driven serverless components with Azure Functions, triggers, and bindings
- Connect services together with queues and messaging instead of brittle direct calls
- Monitor, scale, and troubleshoot applications with Application Insights

Course Outline

Day one: applications on App Service

- Choosing Compute on Azure
 - The Azure compute spectrum: VMs, App Service, Container Apps, Functions, AKS
 - What a managed platform takes off your plate, and what it takes away

- Lab: map three real workloads to the right Azure compute service and defend the choices
- Deploying to App Service
 - App Service plans and tiers, and what they actually buy you
 - Deployment options: from GitHub, from CI, from containers
 - Deployment slots and zero-downtime releases
 - Lab: deploy a web application to App Service and release an update through a staging slot
- Configuration and Secrets
 - App settings, connection strings, and per-slot configuration
 - Managed identity: eliminating credentials from your code
 - Key Vault for the secrets that remain
 - Lab: move configuration and secrets out of code using app settings, managed identity, and Key Vault

Day two: serverless, integration, and operations

- Azure Functions Fundamentals
 - The serverless model: triggers, bindings, and consumption pricing
 - When Functions fit, and when they do not
 - Lab: build an HTTP-triggered function and a queue-triggered function
- Integrating Services
 - Connecting App Service and Functions to storage, queues, and databases
 - Durable Functions for workflows, briefly and honestly
 - Lab: build a small pipeline where a web app hands work to a function through a queue
- Operating What You Built
 - Application Insights: telemetry, distributed traces, and finding the slow part
 - Scaling: autoscale rules, cold starts, and their costs
 - Diagnosing failures: logs, health checks, and common production issues
 - Lab: instrument the application, break it deliberately, and diagnose it from telemetry

Extended Version

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

- API Management: fronting your services properly
- Networking for App Service and Functions: private endpoints and VNet integration
- Deeper CI/CD for the platform, connecting to the practices in *CI/CD with GitHub Actions*
- A capstone: build, secure, and deploy a complete small application on App Service and Functions, with telemetry proving it works