

Developing Solutions for Microsoft Azure (AZ-204T00)

Duration 5 Days

COURSE DESCRIPTION

This course teaches developers how to create end-to-end solutions in Microsoft Azure. Students will learn how to implement Azure compute solutions, create Azure Functions, implement and manage web apps, develop solutions utilizing Azure storage, implement authentication and authorization, and secure their solutions by using KeyVault and Managed Identities. Students will also learn how to connect to and consume Azure services and third-party services, and include event- and message-based models in their solutions. The course also covers monitoring, troubleshooting, and optimizing Azure solutions.

COURSE OBJECTIVES

In this course, you will learn how to:

- Describe Azure App Service key components and value.
- Explain how Azure App Service manages authentication and authorization.
- Identify methods to control inbound and outbound traffic to your web app.
- Deploy an app to App Service using Azure CLI commands.
- Create application settings that are bound to deployment slots.
- Explain the options for installing SSL/TLS certificates for your app.
- Enable diagnostic logging for your app to aid in monitoring and debugging.
- Create virtual app to directory mappings.
- Identify scenarios for which autoscaling is an appropriate solution.
- Create autoscaling rules for a web app.
- Monitor the effects of autoscaling.
- Describe the benefits of using deployment slots.
- Understand how slot swapping operates in App Service.
- Perform manual swaps and enable auto swap.
- Route traffic manually and automatically.
- Explain functional differences between Azure Functions, Azure Logic Apps, and WebJobs
- Describe Azure Functions hosting plan options
- Describe how Azure Functions scale to meet business needs
- Explain the key components of a function and how they are structured
- Create triggers and bindings to control when a function runs and where the output is directed
- Connect a function to services in Azure
- Create a function by using Visual Studio Code and the Azure Functions Core Tools
- Describe the app patterns typically used in durable functions
- Describe the four durable function types
- Explain the function Task Hubs perform in durable functions
- Describe the use of durable orchestrations, timers, and events
- Identify the different types of storage accounts and the resource hierarchy for blob storage.

- Explain how data is securely stored and protected through redundancy.
- Create a block blob storage account by using the Azure Cloud Shell.
- Describe how each of the access tiers are optimized.
- Create and implement a lifecycle policy.
- Rehydrate blob data stored in an archive tier.
- Create an application to create and manipulate data by using the Azure Storage client library for Blob storage.
- Manage container properties and metadata by using .NET and REST.
- Identify the key benefits provided by Azure Cosmos DB
- Describe the elements in an Azure Cosmos DB account and how they are organized
- Explain the different consistency levels and choose the correct one for your project
- Explore the APIs supported in Azure Cosmos DB and choose the appropriate API for your solution
- Describe how request units impact costs
- Create Azure Cosmos DB resources by using the Azure portal.
- Describe the differences between logical and physical partitions
- Choose the appropriate partition key for your solution
- Create a synthetic partition key
- Identify classes and methods used to create resources
- Create resources by using the Azure Cosmos DB .NET v3 SDK
- Write stored procedures, triggers, and user-defined functions by using JavaScript
- Describe the design considerations for creating a virtual machine to support your apps needs.
- Explain the different availability options for Azure VMs.
- Describe the VM sizing options.
- Create an Azure VM by using the Azure CLI.
- Describe what role Azure Resource Manager (ARM) has in Azure and the benefits of using ARM templates.
- Explain what happens when ARM templates are deployed and how to structure them to support your solution.
- Create a template with conditional resource deployments.
- Choose the correct deployment mode for your solution.
- Create and deploy an ARM template by using Visual Studio Code.
- Explain the features and benefits Azure Container Registry offers.
- Describe how to use ACR Tasks to automate builds and deployments.
- Explain the elements in a Dockerfile.
- Build and run an image in the ACR by using Azure CLI.
- Describe the benefits of Azure Container Instances and how resources are grouped.
- Deploy a container instance in Azure by using the Azure CLI.
- Start and stop containers using policies.
- Set environment variables in your container instances.
- Mount file shares in your container instances.
- Identify the components of the Microsoft identity platform.
- Describe the three types of service principals and how they relate to application objects.
- Explain how permissions and user consent operate, and how conditional access impacts your application.



- Explain the benefits of using Microsoft Authentication Library and the application types and scenarios it supports.
- Instantiate both public and confidential client apps from code.
- Register an app with the Microsoft identity platform.
- Create an app that retrieves a token by using the MSAL.NET library.
- Identify the three types of shared access signatures.
- Explain when to implement shared access signatures.
- Create a stored access policy.
- Explain the benefits of using Microsoft Graph.
- Perform operations on Microsoft Graph by using REST and SDKs.
- Apply best practices to help your applications get the most out of Microsoft Graph.
- Describe the benefits of using Azure Key Vault
- Explain how to authenticate to Azure Key Vault
- Set and retrieve a secret from Azure Key Vault by using the Azure CLI
- Explain the differences between the two types of managed identities
- Describe the flows for user- and system-assigned managed identities
- Configure managed identities
- Acquire access tokens by using REST and code
- Explain the benefits of using Azure App Configuration
- Describe how Azure App Configuration stores information
- Implement feature management
- Securely access your app configuration information
- Describe the components (and their functions) of the API Management service.
- Explain how API gateways can help manage calls to your APIs.
- Secure access to APIs by using subscriptions and certificates.
- Create a backend API.
- Describe how Event Grid operates and how it connects to services and event handlers.
- Explain how Event Grid delivers events and how it handles errors.
- Implement authentication and authorization.
- Route custom events to web endpoint by using Azure CLI.
- Describe the benefits of using Event Hubs and how it captures streaming data.
- Explain how to process events.
- Perform common operations with the Event Hubs client library.
- Choose the appropriate queue mechanism for your solution.
- Explain how the messaging entities that form the core capabilities of Service Bus operate.
- Send and receive message from a Service Bus queue by using .NET.
- Identify the key components of Azure Queue Storage
- Create queues and manage messages in Azure Queue Storage by using .NET.
- Explain how Azure Monitor operates as the center of monitoring in Azure.
- Describe how Application Insights works and how it collects events and metrics.
- Instrument an app for monitoring, perform availability tests, and use Application Map to help you monitor performance and troubleshoot issues.
- Explain the key scenarios Azure Cache for Redis covers and its service tiers.
- Identify the key parameters for creating an Azure Cache for Redis instance and interact with the cache.

- Connect an app to Azure Cache for Redis by using .NET Core.
- Explain how the Azure Content Delivery Network works and how it can improve the user experience.
- Control caching behavior and purge content.
- Perform actions on Azure CDN by using the Azure CDN Library for .NET.

COURSE OUTLINE

Module 1. Explore Azure App Service

Module 2. Configure web app settings

Module 3. Scale apps in Azure App Service

Module 4. Explore Azure App Service deployment slots

Module 5. Explore Azure Functions

Module 6. Develop Azure Functions

Module 7. Implement Durable Functions

Module 8. Explore Azure Blob storage

Module 9. Manage the Azure Blob storage lifecycle

Module 10. Work with Azure Blob storage

Module 11. Explore Azure Cosmos DB

Module 12. Implement partitioning in Azure Cosmos DB

Module 13. Work with Azure Cosmos DB

Module 14. Provision virtual machines in Azure

Module 15. Create and deploy Azure Resource Manager templates

Module 16. Manage container images in Azure Container Registry

Module 17. Run container images in Azure Container Instances

Module 18. Explore the Microsoft identity platform

Module 19. Implement authentication by using the Microsoft Authentication Library

Module 20. Implement shared access signatures

Module 21. Explore Microsoft Graph

Module 22. Implement Azure Key Vault

Module 23. Implement managed identities

Module 24. Implement Azure App Configuration

Module 25. Explore API Management

Module 26. Explore Azure Event Grid

Module 27. Explore Azure Event Hubs

Module 28. Discover Azure message queues

Module 29. Monitor app performance

Module 30. Develop for Azure Cache for Redis

Module 31: Develop for storage on CDNs

WHO SHOULD ATTEND

Students in this course are interested in Azure development or in passing the Microsoft Azure Developer Associate certification exam.

PREREQUISITES

To be successful in this course, learners should have the following.

- Hands-on experience with Azure IaaS and PaaS solutions, and the Azure Portal.
- Experience writing in an Azure supported language at the intermediate level. (C#, JavaScript, Python, or Java)
- Ability to write code to connect and perform operations on, a SQL or NoSQL database product. (SQL Server, Oracle, MongoDB, Cassandra or similar)
- Experience writing code to handle authentication, authorization, and other security principles at the intermediate level.
- A general understanding of HTML, the HTTP protocol and REST API interfaces.

Certification

Microsoft Certified: Azure Developer Associate