

AZ-204T00: Developing Solutions for Microsoft Azure

Course Summary

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.

Objectives

At the end of this course, students will be able to:

- Explore Azure App Service
- Configure web app settings
- Scale apps in Azure App Service
- Explore Azure App Service deployment slots
- Explore Azure Functions
- Develop Azure Functions
- Explore Azure Blob storage
- Manage the Azure Blob storage lifecycle
- Work with Azure Blob storage
- Explore Azure Cosmos DB
- Implement partitioning in Azure Cosmos DB
- Work with Azure Cosmos DB
- Manage container images in Azure Container Registry
- Run container images in Azure Container Instances
- Implement Azure Container Apps
- Explore the Microsoft identity platform
- Implement authentication by using the Microsoft Authentication Library
- Implement shared access signatures
- Explore Microsoft Graph
- Implement Azure Key Vault
- Implement managed identities
- Implement Azure App Configuration
- Explore API Management
- Explore Azure Event Grid
- Explore Azure Event Hubs
- Discover Azure message queues
- Monitor app performance
- Develop for Azure Cache for Redis
- Develop for storage on CDNs

Topics

- Explore Azure App Service
- Configure web app settings
- Scale apps in Azure App Service
- Explore Azure App Service deployment slots
- Explore Azure Functions
- Develop Azure Functions
- Explore Azure Blob storage
- Manage the Azure Blob storage lifecycle
- Work with Azure Blob storage
- Explore Azure Cosmos DB
- Work with Azure Cosmos DB
- Manage container images in Azure Container Registry
- Run container images in Azure Container Instances
- Implement Azure Container Apps
- Explore the Microsoft identity platform
- Implement authentication by using the Microsoft Authentication Library
- Implement shared access signatures
- Explore Microsoft Graph
- Implement Azure Key Vault
- Implement managed identities
- Implement Azure App Configuration
- Explore API Management
- Explore Azure Event Grid
- Explore Azure Event Hubs
- Discover Azure message queues
- Monitor app performance
- Develop for Azure Cache for Redis
- Develop for storage on CDNs

AZ-204T00: Developing Solutions for Microsoft Azure

Course Summary (cont'd)

Audience

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.

Duration

Five days

AZ-204T00: Developing Solutions for Microsoft Azure

Course Outline

I. *Explore Azure App Service*

- A. Describe Azure App Service key components and value.
- B. Explain how Azure App Service manages authentication and authorization.
- C. Identify methods to control inbound and outbound traffic to your web app.
- D. Deploy an app to App Service using Azure CLI commands.

II. *Configure web app settings*

- A. Create application settings that are bound to deployment slots.
- B. Explain the options for installing SSL/TLS certificates for your app.
- C. Enable diagnostic logging for your app to aid in monitoring and debugging.
- D. Create virtual app to directory mappings.

III. *Scale apps in Azure App Service*

- A. Identify scenarios for which autoscaling is an appropriate solution.
- B. Create autoscaling rules for a web app.
- C. Monitor the effects of autoscaling.

IV. *Explore Azure App Service deployment slots*

- A. Describe the benefits of using deployment slots.
- B. Understand how slot swapping operates in App Service.
- C. Perform manual swaps and enable auto swap.
- D. Route traffic manually and automatically.

V. *Explore Azure Functions*

- A. Explain functional differences between Azure Functions, Azure Logic Apps, and WebJobs
- B. Describe Azure Functions hosting plan options
- C. Describe how Azure Functions scale to meet business needs

VI. *Develop Azure Functions*

- A. Explain the key components of a function and how they are structured

- B. Create triggers and bindings to control when a function runs and where the output is directed
- C. Connect a function to services in Azure
- D. Create a function by using Visual Studio Code and the Azure Functions Core Tools

VII. *Explore Azure Blob storage*

- A. Identify the different types of storage accounts and the resource hierarchy for blob storage.
- B. Explain how data is securely stored and protected through redundancy.
- C. Create a block blob storage account by using the Azure Cloud Shell.

VIII. *Manage the Azure Blob storage lifecycle*

- A. Describe how each of the access tiers are optimized.
- B. Create and implement a lifecycle policy.
- C. Rehydrate blob data stored in an archive tier.

IX. *Work with Azure Blob storage*

- A. Create an application to create and manipulate data by using the Azure Storage client library for Blob storage.
- B. Manage container properties and metadata by using .NET and REST.

X. *Explore Azure Cosmos DB*

- A. Identify the key benefits provided by Azure Cosmos DB
- B. Describe the elements in an Azure Cosmos DB account and how they are organized
- C. Explain the different consistency levels and choose the correct one for your project
- D. Explore the APIs supported in Azure Cosmos DB and choose the appropriate API for your solution
- E. Describe how request units impact costs
- F. Create Azure Cosmos DB resources by using the Azure portal.

AZ-204T00: Developing Solutions for Microsoft Azure

Course Outline (cont'd)

XI. *Work with Azure Cosmos DB*

- A. Identify classes and methods used to create resources
- B. Create resources by using the Azure Cosmos DB .NET v3 SDK
- C. Write stored procedures, triggers, and user-defined functions by using JavaScript

XII. *Manage container images in Azure Container Registry*

- A. Explain the features and benefits Azure Container Registry offers.
- B. Describe how to use ACR Tasks to automate builds and deployments.
- C. Explain the elements in a Dockerfile.
- D. Build and run an image in the ACR by using Azure CLI.

XIII. *Run container images in Azure Container Instances*

- A. Describe the benefits of Azure Container Instances and how resources are grouped.
- B. Deploy a container instance in Azure by using the Azure CLI.
- C. Start and stop containers using policies.
- D. Set environment variables in your container instances.
- E. Mount file shares in your container instances.

XIV. *Implement Azure Container Apps*

- A. Describe the benefits of Azure Container Instances and how resources are grouped
- B. Deploy a container instance in Azure by using the Azure CLI
- C. Start and stop containers using policies
- D. Set environment variables in your container instances
- E. Mount file shares in your container instances

XV. *Explore the Microsoft identity platform*

- A. Identify the components of the Microsoft identity platform.
- B.

- C. Describe the three types of service principals and how they relate to application objects.
- D. Explain how permissions and user consent operate, and how conditional access impacts your application.

XVI. *Implement authentication by using the Microsoft Authentication Library*

- A. Explain the benefits of using Microsoft Authentication Library and the application types and scenarios it supports.
- B. Instantiate both public and confidential client apps from code.
- C. Register an app with the Microsoft identity platform.
- D. Create an app that retrieves a token by using the MSAL.NET library.

XVII. *Implement shared access signatures*

- A. Identify the three types of shared access signatures.
- B. Explain when to implement shared access signatures.
- C. Create a stored access policy.

XVIII. *Explore Microsoft Graph*

- A. Explain the benefits of using Microsoft Graph.
- B. Perform operations on Microsoft Graph by using REST and SDKs.
- C. Apply best practices to help your applications get the most out of Microsoft Graph.

XIX. *Implement Azure Key Vault*

- A. Describe the benefits of using Azure Key Vault
- B. Explain how to authenticate to Azure Key Vault
- C. Set and retrieve a secret from Azure Key Vault by using the Azure CLI

AZ-204T00: Developing Solutions for Microsoft Azure

Course Outline (cont'd)

XX. *Implement managed identities*

- A. Explain the differences between the two types of managed identities
- B. Describe the flows for user- and system-assigned managed identities
- C. Configure managed identities
- D. Acquire access tokens by using REST and code

XXI. *Implement Azure App Configuration*

- A. Explain the benefits of using Azure App Configuration
- B. Describe how Azure App Configuration stores information
- C. Implement feature management
- D. Securely access your app configuration information

XXII. *Explore API Management*

- A. Describe the components (and their functions) of the API Management service.
- B. Explain how API gateways can help manage calls to your APIs.
- C. Secure access to APIs by using subscriptions and certificates.
- D. Create a backend API.

XXIII. *Explore Azure Event Grid*

- A. Describe how Event Grid operates and how it connects to services and event handlers.
- B. Explain how Event Grid delivers events and how it handles errors.
- C. Implement authentication and authorization.
- D. Route custom events to web endpoint by using Azure CLI.

XXIV. *Explore Azure Event Hubs*

- A. Describe the benefits of using Event Hubs and how it captures streaming data.
- B. Explain how to process events.
- C. Perform common operations with the Event Hubs client library.

XXV. *Discover Azure message queues*

- A. Choose the appropriate queue mechanism for your solution.
- B. Explain how the messaging entities that form the core capabilities of Service Bus operate.
- C. Send and receive message from a Service Bus queue by using .NET.
- D. Identify the key components of Azure Queue Storage
- E. Create queues and manage messages in Azure Queue Storage by using .NET.

XXVI. *Monitor app performance*

- A. Explain how Azure Monitor operates as the center of monitoring in Azure.
- B. Describe how Application Insights works and how it collects events and metrics.
- C. Instrument an app for monitoring, perform availability tests, and use Application Map to help you monitor performance and troubleshoot issues.

XXVII. *Develop for Azure Cache for Redis*

- A. Explain the key scenarios Azure Cache for Redis covers and its service tiers.
- B. Identify the key parameters for creating an Azure Cache for Redis instance and interact with the cache.
- C. Connect an app to Azure Cache for Redis by using .NET Core.

XXVIII. *Develop for storage on CDNs*

- A. Explain how the Azure Content Delivery Network works and how it can improve the user experience.
- B. Control caching behavior and purge content.
- C. Perform actions on Azure CDN by using the Azure CDN Library for .NET.