



Developing Solutions for Microsoft Azure

THIRD EDITION

Exam Ref AZ-204

Santiago Fernández Muñoz

FREE SAMPLE CHAPTER |



Exam Ref AZ-204 Developing Solutions for Microsoft Azure, Third Edition

Santiago Fernández Muñoz

Exam Ref AZ-204 Developing Solutions for Microsoft Azure, Third Edition

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

Copyright © 2025 by Pearson Education, Inc.

Hoboken, New Jersey

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 9780138312138

ISBN-10: 0138312133

Library of Congress Control Number: 2024938412

\$PrintCode

TRADEMARKS

Microsoft and the trademarks listed at microsoft.com on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

WARNING AND DISCLAIMER

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

SPECIAL SALES

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

CREDITS

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

ASSOCIATE EDITOR
Shourav Bose

DEVELOPMENT EDITOR
Songlin Qiu

MANAGING EDITOR
Sandra Schroeder

SENIOR PROJECT EDITOR
Tracey Croom

COPY EDITOR
Dan Foster

INDEXER
Timothy Wright

PROOFREADER
Barbara Mack

TECHNICAL EDITOR
Doug Holland

EDITORIAL ASSISTANT
Cindy Teeters

COVER DESIGNER
Twist Creative, Seattle

COMPOSITOR
codeMantra

GRAPHICS
codeMantra

Contents at a glance

	<i>About the Author</i>	<i>ix</i>
	<i>Introduction</i>	<i>x</i>
CHAPTER 1	Develop an Azure Infrastructure as a Service solution	1
CHAPTER 2	Develop for Azure storage	67
CHAPTER 3	Implement Azure security	117
CHAPTER 4	Monitor, troubleshoot, and optimize Azure solutions	189
CHAPTER 5	Connect to and consume Azure services and third-party services	223
CHAPTER 6	AZ-204 developing solutions for Microsoft Azure exam updates	261
	<i>Index</i>	<i>265</i>

Contents

Introduction	x
<i>Organization of this book</i>	<i>x</i>
<i>Preparing for the exam</i>	<i>x</i>
<i>Microsoft certifications</i>	<i>xi</i>
<i>Access the Exam Updates chapter and online references</i>	<i>xi</i>
<i>Errata, updates, & book support</i>	<i>xii</i>
<i>Stay in touch</i>	<i>xii</i>
Chapter 1 Develop an Azure Infrastructure as a Service solution	1
Skill 1.1: Implement containerized solutions	2
Create and manage container images for solutions	2
Publish an image to the Azure Container Registry	8
Run containers by using an Azure Container Instance	12
Create solutions by using Azure Container Apps	14
Skill 1.2: Implement Azure App Service web apps	20
Create an Azure App Service web app	21
Enable diagnostics logging	25
Deploy code to a web app	28
Configure web app settings including Secure Sockets Layer, API settings, and connection strings	31
Implement autoscaling rules, including scheduled autoscaling, and scaling by operational or system metrics	34
Configure deployment slots	39
Skill 1.3: Implement Azure Functions	43
Create and configure an Azure Function App	44
Implement input and output bindings for a function	47
Implement function triggers by using data operations, timers, and webhooks	53
Chapter summary	63
Thought experiment	65
Thought experiment answers	66

Chapter 2	Develop for Azure storage	67
	Skill 2.1: Develop solutions that use Cosmos DB storage	67
	Perform operations on containers and items by using the SDK	68
	Set the appropriate consistency level for operations	86
	Implement change feed notifications	89
	Skill 2.2: Develop solutions that use Blob Storage	93
	Set and retrieve properties and metadata	94
	Interact with data using the appropriate SDK	99
	Implement storage policies and data life cycle management	106
	Implement static site hosting	113
	Chapter summary	115
	Thought experiment	116
	Thought experiment answers	116
Chapter 3	Implement Azure security	117
	Skill 3.1: Implement user authentication and authorization	117
	Authenticate and authorize users by using the Microsoft Identity platform	118
	Authenticate and authorize users and apps by using Microsoft Entra ID	141
	Create and implement shared access signatures	146
	Implement solutions that interact with Microsoft Graph	158
	Skill 3.2: Implement secure cloud solutions	164
	Secure app configuration data by using App Configuration or Azure Key Vault	164
	Develop code that uses keys, secrets, and certificates stored in Azure Key Vault	172
	Implement Managed Identities for Azure resources	179
	Chapter summary	184
	Thought experiment	186
	Thought experiment answers	187

Chapter 4	Monitor, troubleshoot, and optimize Azure solutions	189
	Skill 4.1: Implement caching for solutions	189
	Configure cache and expiration policies for Azure	
	Cache for Redis	190
	Implement secure and optimized application cache patterns including data sizing, connections, encryption, and expiration	193
	Implement Azure Content Delivery Network endpoints and profiles	200
	Skill 4.2: Troubleshoot solutions by using Application Insights	204
	Configure an app or service to use Application Insights	204
	Monitor and analyze metrics, logs, and traces	212
	Implement Application Insights Web Test and Alerts	216
	Chapter summary	219
	Thought experiment	220
	Thought experiment answers	220
Chapter 5	Connect to and consume Azure services and third-party services	223
	Skill 5.1: Implement API Management	223
	Create an APIM instance	224
	Create and document APIs	225
	Configure authentication for APIs	229
	Implement policies for APIs	231
	Skill 5.2: Develop event-based solutions	234
	Implement solutions that use Azure Event Grid	234
	Implement solutions that use Azure Event Hub	242
	Skill 5.3: Develop message-based solutions	248
	Implement solutions that use Azure Service Bus	249
	Implement solutions that use Azure Queue Storage	255
	Chapter summary	259
	Thought experiment	259
	Thought experiment answers	260

Chapter 6	AZ-204 developing solutions for Microsoft Azure exam updates	261
	The purpose of this chapter	261
	About possible exam updates	262
	Impact on you and your study plan	262
	News and commentary about the exam objective updates.....	262
	Updated technical content	263
	Objective mapping	263
	 <i>Index</i>	 265

About the Author

SANTIAGO FERNÁNDEZ MUÑOZ is a Senior Solution Architect involved in developing international projects. He started his career as a Systems Engineer and jumped into the professional development world, attracted by the possibilities and complexities of distributed computing. He runs his own company, mainly focused on providing services to the industrial environment, specializing in photovoltaic environments and industrial cybersecurity.

Introduction

Most books take a very low-level approach, teaching you how to use individual classes and accomplish fine-grained tasks. Through this book, we review the main technologies that Microsoft offers for deploying different kinds of solutions into Azure. From the most classic and conservative approaches using Azure Virtual Machines to the latest technologies implementing event-based or message-based patterns with Azure Event Grid or Azure Service Bus, this book reviews the basics for developing most types of solutions using Azure services. The book also provides code examples that illustrate how to implement most of the concepts covered. This book should be used as an introduction to implementing more complex solutions. Although the book covers some basic concepts, you should have basic programming experience using ASP.NET, .NET Framework, or .NET Core, as well as using Git.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the “Need more review?” links you’ll find in the text to find more information and take the time to research and study the topic. Great information is available on MSDN, TechNet, and in blogs and forums.

Organization of this book

This book is organized by the “Skills measured” list published for the exam. The “Skills measured” list is available for each exam on the Microsoft Learn website: learn.microsoft.com/en-us/credentials/certifications/resources/study-guides/az-204. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in each topic area determine a chapter’s organization. If an exam covers six major topic areas, for example, the book will contain six chapters.

Preparing for the exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. This book is not designed to teach you new skills.

We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the *Exam Ref* and another study guide for your at-home preparation and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you. Learn more about available classroom training, online courses, and live events at microsoft.com/learn.

Note that this *Exam Ref* is based on publicly available information about the exam and the author's experience. To safeguard the integrity of the exam, authors do not have access to the live exam.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

NEED MORE REVIEW? ALL MICROSOFT CERTIFICATIONS

For information about Microsoft certifications, including a full list of available certifications, go to www.microsoft.com/learn.

Access the Exam Updates chapter and online references

The final chapter of this book, "AZ-204 developing solutions for Microsoft Azure exam updates," will be used to provide information about new content per new exam topics, content that has been removed from the exam objectives, and revised mapping of exam objectives to chapter content. The chapter will be made available from the link below as exam updates are released.

Throughout this book are addresses to webpages that the author has recommended you visit for more information. Some of these links can be very long and painstaking to type, so we've shortened them for you to make them easier to visit. We've also compiled them into a single list that readers of the print edition can refer to while they read.

The URLs are organized by chapter and heading. Every time you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

Download the Exam Updates chapter and the URL list at MicrosoftPressStore.com/ERAZ2043e/downloads.

Errata, updates & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

MicrosoftPressStore.com/ERAZ2043e/errata

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit *MicrosoftPressStore.com/Support*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *support.microsoft.com*.

Stay in touch

Let's keep the conversation going! We're on Twitter: *twitter.com/MicrosoftPress*.

Develop for Azure storage

All applications work with information or data. Applications create, transform, model, or operate with that information. Regardless of the type or volume of data that your application uses, eventually you need to save it persistently so that it can be used later.

Storing data is not a simple task, and designing storage systems for that purpose is even more complicated. Perhaps your application must deal with terabytes or even petabytes of information, or you might work with an application that needs to be accessed from different countries, and you need to minimize the time required to access it. Cost efficiency is also a requirement in any project. In general, many requirements make designing and maintaining storage systems difficult.

Microsoft Azure offers different storage solutions in the cloud to satisfy your application storage requirements. Azure offers solutions for making your storage cost-effective and for minimizing latency.

Skills covered in this chapter:

- Skill 2.1: Develop solutions that use Cosmos DB storage
- Skill 2.2: Develop solutions that use Blob Storage

Skill 2.1: Develop solutions that use Cosmos DB storage

Cosmos DB is a premium storage service that Azure provides for satisfying your need for a globally distributed, low-latency, highly responsive, and always-online database service. Cosmos DB has been designed with scalability and throughput in mind. One of the most significant differences between Cosmos DB and other storage services offered by Azure is how easily you can scale your Cosmos DB solution across the globe by merely clicking a button and adding a new region to your database.

Another essential feature that you should consider when evaluating this type of storage service is how you can access this service from your code and how hard it would be to migrate your existing code to a Cosmos DB-based storage solution. The good news is that Cosmos DB offers different APIs for accessing the service. The best API for you depends on the type of data that you want to store in your Cosmos DB database. You store your data using Key-Value, Column-Family, Documents, or Graph approaches. Each of the different

APIs that Cosmos DB offers allows you to store your data with different schemas. Currently, you can access Cosmos DB using SQL, Cassandra, Table, Gremlin, and MongoDB APIs.

This skill covers how to

- Perform operations on containers and items by using the SDK
- Set the appropriate consistency level for operations
- Implement change feed notifications

Perform operations on containers and items by using the SDK

When working with Cosmos DB, you have several layers in the hierarchy of entities managed by the Cosmos DB account. The first layer is the Azure Cosmos DB account, where you choose the API you want to use to access your data. Remember that this API has implications for how the data is stored in the databases.

The second layer in the hierarchy is the database. You can create as many databases as you need in your Cosmos DB account. Databases are a way of grouping containers; you can think of databases like namespaces. At this level, you can configure the throughput associated with the containers included in the database.

When planning how to store the information that your application needs to work, you must consider the structure you need to use for storing that information. You may find that some parts of your application need to store information using a key-value structure. In contrast, others may need a more flexible, schema-less structure in which you save the information into documents. One fundamental characteristic of your application might be that you need to store the relationship between entities and use a graph structure for storing your data.

Cosmos DB offers a variety of APIs for storing and accessing your data, depending on the requirements of your application:

- **NoSQL** This is the core and default API for accessing your data in your Cosmos DB account. This core API allows you to query JSON objects using SQL syntax, which means you don't need to learn another query language. Under the hood, the SQL API uses the JavaScript programming model for expression evaluation, function invocations, and typing systems. You use this API when you need to use a data structure based on documents.
- **Table** You can think of the Table API as the evolution of the Azure Table Storage service. This API benefits from the high-performance, low-latency, and high-scalability features of Cosmos DB. You can migrate from your current Azure Table Storage service with no code modification in your application. Another critical difference between Table API for Cosmos DB and Azure Table Storage is that you can define your own indexes in your tables. In the same way you do with the Table Storage service, the Table API allows you to store information in your Cosmos DB account using a data structure based on documents.

- **Cassandra** Cosmos DB implements the wire protocol for the Apache Cassandra database into the options for storing and accessing data in the Cosmos DB database. This allows you to forget about operations and performance-management tasks related to managing Cassandra databases. In most situations, you can migrate your application from your current Cassandra database to Cosmos DB using the Cassandra API by merely changing the connection string. Azure Cosmos DB Cassandra API is compatible with the CQLv4 wire protocol. Cassandra is a column-based database that stores information using a key-value approach.
- **MongoDB** You can access your Cosmos DB account by using the MongoDB API. This NoSQL database allows you to store the information for your application in a document-based structure. Cosmos DB implements the wire protocol compatible with MongoDB 3.2. This means that any MongoDB 3.2 client driver that implements and understands this protocol definition can connect seamlessly with your Cosmos DB database using the MongoDB API.
- **PostgreSQL** This service is built on top of native PostgreSQL, which means you can use your code directly with Azure Cosmos DB for PostgreSQL without any substantial modification. This is a managed service, so Microsoft takes care of all the details regarding performance, availability, geo-replication, and all the features that the Cosmos DB service offers.
- **Gremlin** Based on the Apache TinkerPop graph transversal language or Gremlin, this API allows you to store information in Cosmos DB using a graph structure. This means that instead of storing only entities, you store:
 - **Vertices** You can think of a vertex as an entity in other information structures. In a typical graph structure, a vertex could be a person, a device, or an event.
 - **Edges** These are the relationships between vertices. A person can know another person, a person might own a type of device, or a person might attend an event.
 - **Properties** These are attributes you can assign to a vertex or an edge.

Beware that you cannot mix these APIs in a single Cosmos DB account. You must define the API you want to use for accessing your Cosmos DB account when creating the account. Once you have created the account, you won't be able to change the API to access it.

Azure offers SDKs for working with the different APIs you can use to connect to Cosmos DB. Supported languages are .NET, Java, Node.js, and Python. Depending on the API you want to use for working with Cosmos DB, you can also use other languages such as Spring Data, Spark V3, or Golang.

NOTE AZURE COSMOS DB EMULATOR

You can use the Azure Cosmos DB emulator during the development stage of your application. You should keep in mind that there are some limitations when working with the emulator instead of a real Cosmos DB account. The emulator is only supported on Windows platforms or Docker for Windows. You can review all characteristics of the Cosmos DB emulator at <https://docs.microsoft.com/en-us/azure/cosmos-db/local-emulator>.

Index

A

- access tiers, Azure Blob Storage, 106–107, 109–110
- account
 - Admin, enabling, 12
 - Azure Queue Storage, creating, 227–256
 - Cosmos DB, 68
 - connection string, 92–93
 - creating a new collection, 75–77
 - creating and editing elements, 78
 - Microsoft 365, 158–159
- SAS
 - policy, 150
 - URI parameters, 148–149
 - Storage, interact with data using the appropriate SDK, 99–106
- ACI (Azure Container Instance)
 - creating from a container registry, 12–14
 - running a container, 12–14
- ACR (Azure Container Registry)
 - pricing tiers, 9
 - publishing an image, 8–9
 - push the image to the registry, 9–10
 - tag the image, 9
- Admin account, enabling, 12
- alert, URL ping test, 218–219
- algorithm, maxmemory policy, 193
- API (application programming interface), 223
 - adding to APIM instance
 - using no specification, 226–227
 - using OpenAI specification, 225–226
 - associating with a product, 228
 - Cosmos DB, 67–68
 - Cassandra, 68
 - Gremlin, 68–69
 - MongoDB, 68
 - NoSQL, 68, 78–82, 85
 - PostgreSQL, 68
 - selecting, 75
 - Table, 68
 - KeyVault, 172–174, 177–179
 - Kubernetes, 14
 - Library, applying transformations, 232–233
 - REST, 224
 - revisions and versions, 229
 - SOAP, 224
 - subscription, 229
 - creating, 229–230
 - scope, 229
 - testing, 232
- APIM (Azure API Management) service, 224
 - associating an API with a product, 228
 - developer portal, 227–228
 - instance
 - adding back-end API using OpenAI specification, 225–226
 - adding back-end API without using a specification, 226–227
 - creating, 224–225
 - policy/ies, 233–234
 - creating, 231
 - scope, 229
 - request/responses, 231
 - subscription, 229
 - creating, 229–230
 - scope, 229
 - version 2, 224
- application
 - authentication flow, 159
 - configuring to use Application Insights, 204–206
 - console, 189
 - creating, 251–252, 256–257
 - Event Processor, 246–247
 - listing all users in your tenant, 161
 - registering in Microsoft Entra ID, 160–161
 - lifetime, 16
 - logging, 26–27
 - monitoring, 189

application, continued

- application, *continued*
 - user experience, 189
 - web
 - application owner, 144–145
 - authentication, 22, 121–122
 - authentication, Microsoft Entra ID, 143–144
 - authorization, 146
 - authorization server, creating, 124–134
 - Azure Web App, creating, 200–202
 - caching, 189–190
 - creating, 121
 - deploying code to a, 28–30
 - registering in Microsoft Entra ID, 142–143
 - settings, 31–33
 - settings, authentication, 142–143
 - settings, Certificates and Secrets, 143
 - URL ping test, 216–219
- Application Insights
 - accessing from Visual Studio and Azure portal, 208
 - adding custom events and metrics, 208, 210
 - configuring on an app, 204–206
 - instrumentation, 205–206
 - instrumentation libraries, 206
 - messages, 210
 - telemetry, 205, 206–207
 - workplace schema, 214–215
- app/s. *See also* application; web application
 - Azure Functions, creating, 44–47
 - container
 - creating, 16–17
 - scale rule, 17–18
- ASP.NET, accessing an app setting, 33
- authentication, 118, 141, 144–145
 - API (application programming interface), subscription, 229–230
 - client credentials flow, 163
 - Cosmos DB Client, 82
 - flow, 159, 163
 - form-based, 118
 - Microsoft Entra ID, 141
 - application owner, 144–145
 - registering your web application, 142–143
 - web application authentication, 143–144
 - multifactor, 117
 - OAuth
 - basic authentication flow, 123–124
 - creating a client application, 136–139
 - creating a resource server, 134–136
 - creating an authorization server, 124–134
 - roles, 122–123
 - testing the application, 140
 - OIDC (OpenID Connect), 120, 141–142
 - passwordless, 82
 - stateful, 118
 - token-based, 119–120
 - web application, 22, 121–122
 - Microsoft Entra ID, 143–144
 - settings, 142–143
 - authorization, 117, 141. *See also* OAuth
 - server, creating, 124–134
 - shared key, 147
 - web application, 146
 - AuthorizationPermissionMismatch error code, 155
 - autoscaling, 34–35
 - patterns, 38
 - profile condition, 36–37
 - rule, 35, 36–38
 - az webapp log tail command, 27
 - AZ-204 Developing Solutions for Microsoft Azure exam
 - objective mapping, 263–264
 - updates, 261–263
 - Azure App Configuration, 164–165
 - store, 167
 - creating, 165–166
 - key-value pair, 166, 170–171
 - pricing tiers, 165
 - Azure App Service, 20. *See also* web application
 - App Service plan, 21–23
 - deployment slots, 28, 30, 39–43
 - diagnostic logging, 25–27
 - granting access to your application in Azure Key Vault, 183–184
 - hybrid connections, 22
 - pricing tiers, 25
 - Resource Group, creating, 23–24
 - settings, 165
 - VM (virtual machines), 21
 - VNet integration, 22
 - web application, authentication methods, 22
 - Azure Blob Storage, 93
 - access tiers, 106–107, 109–110
 - implement storage policies, 106–113
 - implementing static site hosting, 113–114
 - interact with data using the appropriate SDK, 99–106
 - life cycle management policy, 107–109
 - pricing, 107
 - set and retrieve properties and metadata, 94–98

- Azure CDN, 200, 202
 - Azure Container Apps, 14, 15
 - Azure Cosmos DB emulator, 69
 - Azure Cosmos DB SDK, 82
 - Azure Event Grid, 234–235, 242
 - processing events, 238–240
 - topic
 - creating, 235–236
 - publishing events to, 236–238
 - verifying, 241
 - Azure Event Hub, 242–243
 - consumer groups, 243
 - event publisher, 242
 - hub
 - creating, 243–244
 - sending and consuming events from, 244–245
 - partition, 242–243
 - Azure Functions, 43–44. *See also* function app, creating, 44–47
 - connecting to Event Grid Topic, 238–240
 - monitoring, 241
 - versions, 56
 - Azure Key Vault, 164, 165, 172
 - API, 172–174, 177–179
 - creating, deleting, updating, and reading items, 174–179
 - granting access to Azure App Service application, 183–184
 - Azure Monitor, 212
 - Log Analytics, 214–215
 - reviewing graphs, 212–213
 - Azure Pipeline, 29–30
 - Azure portal
 - creating a new ACI, 12–14
 - enabling the Admin account, 12
 - Azure Queue Storage, 255
 - creating an account, 227–256
 - maximum message size, 255
 - queue, creating, 227–256
 - Azure Redis Cache, 190
 - Cache-Aside, 193–194
 - content caching, 194
 - database
 - connecting, 195–199
 - creating, 191–192
 - data size, 199
 - distributed transactions, 194
 - end-of-life time, 192
 - job and message queuing, 194
 - memory policy, 192–193
 - scaling, 191
 - service tiers, 190–191
 - StackExchange.Redis client, 194
 - timeout values, 199
 - user session caching, 194
 - Azure Repos repository, creating a new repo, 29
 - Azure Service Bus, 249
 - namespace, creating, 249–250
 - topic
 - creating, 250
 - publishing messages to, 251–252
 - Azure Storage, SAS (shared access signatures), 147
 - account, 148–150
 - token, 148
 - user delegation, 148
 - Azure Web App, creating, 200–202
- ## B
- back-end API, adding to APIM instance
 - using no specification, 226–227
 - using OpenAPI specification, 225–226
 - binding/s
 - configuration, 51–53
 - declaring, 48
 - expression, 51–52
 - extensions, 48–49
 - input, 48
 - output, 48, 50–52
 - blob rehydration, 107
 - blob storage, 93
 - access tiers, 106–107, 109–110
 - implement storage policies, 106–113
 - implementing static site hosting, 113–114
 - interact with data using the appropriate SDK, 99–106
 - lease, 103–104
 - life cycle management policy, 107–109
 - service SAS, 151–153
 - set and retrieve properties and metadata, 94–98
 - BlobLeaseClient object, 105
 - breakpoints, 176
 - Build() method, 170

C

- Cache-Aside, 193–194
- caching, 204. *See also* Azure Redis Cache
 - Azure Redis Cache, memory policy, 192–193
 - user session, 194
 - web application, 189–190
- Cassandra API, 68, 88–89
- CDN (Content Delivery Network), 200
 - advanced options, 203–204
 - propagation, 203
- certificates, storing, 177–179
- Certificates and Secrets settings, web application, 143
- change feed notifications, Cosmos DB
 - all versions and deletes change feed mode, 90
 - implementing, 91–92
 - latest version change feed mode, 90
 - pull mode, 90
 - push mode, 90
 - reading, 91
- classes, creating, 84
- Client ID, 180
- cloud/cloud computing
 - building an image in the, 10–11
 - PaaS (Platform as a Service), 1
- CMD instruction, Dockerfile, 7
- code, 2
 - accessing an app setting from ASP.NET, 33
 - accessing an app setting from PHP page, 33
 - adding OAuth authorization server, 126–127
 - adding secret information to the home page, 183
 - AppSettings.cs, 168–170
 - AppSettings.json configuration file, 95–96, 100–101, 110–111
 - appsettings.json file, 237
 - authorization code grant, 139
 - Authorize.cshtml, 133
 - AuthorizeError.cshtml, 134
 - breakpoints, 176
 - Clients.cs, 130
 - Cosmos DB Azure function trigger, 92
 - creating, deleting, updating, and reading Key Vault items, 174–176
 - deploying to a web app, 28–30
 - dotnet add package System.IO, 156–158
 - function
 - bindings, 48–53
 - trigger, 47–48
 - Function1.cs, 239–240, 244–245
 - getting a secret from the key vault, 182
 - HomeController class, 208–210
 - HomeController RedisCache method, 196–197
 - index method in ManageController.cs, 137–138
 - life cycle management policy definition, 108
 - MeController.cs, 135
 - NewItemCreatedEvent.cs, 237, 238
 - OAuthController.cs, 132
 - OnValidateClientRedirectUri delegate, 127–129
 - Paths.cs, 131
 - Program.cs, 154–155, 162, 246–247, 252–253, 253–254, 257–258
 - Program.cs C# class, 96–97, 101–102, 111–112
 - Program.cs extension, 146
 - Program.cs Main method, 237–238
 - Program.cs modification, 104–105
 - RedisCache View, 197–198
 - reuse, 234
 - setting user-defined metadata, 97–98
 - Startup.WebApi.cs, 135–136
- commands
 - az webapp log tail, 27
 - Docker
 - docker build, 4
 - docker container ls, 5, 8
 - docker exec -it, 8
 - docker image ls, 7
 - docker image rm, 8
 - docker pull, 8
 - docker push, 8
 - docker run, 8
 - dotnet new webapi, 4
 - Redis, EXPIRE, 192
 - transaction, 194
- configuration
 - binding, 51–53
 - deployment slot, 26–39
 - trigger, 59–60
- connected registries, 11
- connection string, security, 196
- ConnectionMultiplexer object, 194, 198–199
- connectors, Microsoft Graph, 159
- console application
 - creating, 251–252, 253, 256–257
 - Event Processor, 246–247
 - listing all users in your tenant, 161
 - registering in Microsoft Entra ID, 160–161

- consumer groups, 243
- container/s, 2, 15
 - app
 - creating, 16–17
 - scale rule, 17–18
 - application lifetime, 16
 - Cosmos DB
 - properties, 73–74
 - throughput, 73
 - Docker, 2–3
 - environment, 15
 - groups, 14
 - image, 3, 9
 - building in the cloud, 10–11
 - creating, 3–6
 - publish to the Azure Container Registry, 8–11
 - revision, 15–16
 - run by using an ACI (Azure Container Instance), 12–14
 - scaling, 2
 - schema-agnostic, 73
 - sidecar, 15
 - volume, 3
- content caching, Azure Redis Cache, 194
- continuous deployment, 29–30, 41
- cookie, 118
- COPY instruction, Dockerfile, 6
- Cosmos DB, 67–68
 - account, 68
 - connection string, 92–93
 - creating a new collection, 75–77
 - creating and editing elements, 78
 - API, 67–68
 - Cassandra, 68
 - Gremlin, 68–69
 - MongoDB, 68
 - NoSQL, 68, 78–82, 85
 - PostgreSQL, 68
 - selecting, 75
 - Table, 68
- change feed notifications, 89–90
 - all versions and deletes change feed mode, 90
 - implementing, 91–92
 - latest version change feed mode, 90
 - pull mode, 90
 - push mode, 90
 - reading, 91
- client authentication, 82
- container, 70
 - properties, 73–74
 - throughput, 73
- databases, viewing, 86
- documents
 - creating, 83–84
 - managing, 84
 - querying, 84
- partition, 70
 - hot, 71
 - key, 70
 - key, choosing, 71–72
 - logical, 71
- SDK, 69
- setting the appropriate consistency level for operations, 86–88
 - Cassandra or MongoDB APIs, 88–89
 - NoSQL or Table APIs, 88
 - trade-offs, 88
- trigger, 55–56
- cost, estimated monthly, 76
- creating
 - ACI (Azure Container Instance), 12–14
 - APIM instance, 224–225
 - App Service plan, 22–23
 - Azure CDN profile, 202
 - Azure Function App, 44–47
 - Azure Redis Cache database, 191–192
 - Azure Web App, 200–202
 - client application, 136–139
 - console application, 246, 251–252, 253, 256–257
 - container app, 16–17
 - container image, 3–6
 - deployment slot, 43
 - Dockerfile, 3
 - event hub, 243–244
 - namespace, 249–250
 - policies, 231
 - queue, 227–256
 - resource server, 134–136
 - scale rule, 17–18
 - subscription, 229–230
 - system-assigned managed identity, 180–182
 - topic, 235–236, 250
 - user delegation SAS, 154–155
 - web application, 121, 124

D

- data operation trigger, 54
- database
 - Azure Redis Cache
 - connecting, 195–199
 - creating, 191–192
 - data size, 199
 - Redis as, 194
 - viewing in your Cosmos DB account, 86
- declaring a binding, 48
- DefaultAzureCredential() method, 182–183
- DefaultAzureCredential object, 97
- dependency injection, 210
- deployment slot, Azure App Service, 28, 30
 - configuration, 26–39
 - creating, 43
- developer portal, APIM, 227–228
- diagnostic logging, Azure App Service, 25–27
- Docker, 2–3. *See also* command, Docker
 - Compose, 7
 - Desktop, 4
 - service, 7
- docker build command, 4
- docker container ls command, 5, 8
- docker exec -it command, 8
- docker image ls command, 7
- docker image rm command, 8
- docker pull command, 8
- docker push command, 8
- docker run command, 8
- Dockerfile, 3, 8
 - CMD instruction, 7
 - COPY instruction, 6
 - creating, 3–6
 - ENTRYPOINT instruction, 7
 - ENV instruction, 6–7
 - EXPOSE instruction, 7
 - FROM instruction, 6
 - for production environment, 7
 - RUN instruction, 6
 - WORKDIR instruction, 6
- document/s
 - creating, 83–84
 - managing, 84
 - policy
 - life cycle management, 107–109
 - sections, 231
 - querying, 84

- dotnet new webapi command, 4
- dynamic content, 189

E

- edges, 69
- encryption, key, 164
- endpoint
 - Microsoft Identity platform, 120
 - topic, 235
 - creating, 235–236
 - publishing events to, 236–238
- Entra ID, 141
- ENTRYPOINT instruction, Dockerfile, 7
- ENV instruction, Dockerfile, 6–7
- environment, 15
- estimated monthly cost, 76
- event/s, 235
 - Application Insights, 208, 210
 - driven architecture, 234–235
 - handler, 235, 241
 - hub, 248
 - creating, 243–244
 - processor, 248
 - sending and consuming events from, 244–245
 - processing/processor, 238–240, 246–247
 - publisher, 242
 - retry policy, 242
 - source, 235
 - subscription, 235
 - topic, 235
 - creating, 235–236
 - publishing, 236–238
- EXPIRE command, 192
- EXPOSE instruction, Dockerfile, 7

F

- flow authentication, 163
- Forbidden Access Error, 177
- form-based authentication, 118
- function/s
 - binding/s
 - configuration, 51–53
 - declaring, 48
 - extensions, 48–49

- input, 48, 50–51
- output, 48, 50–51
- trigger, 47–48, 53–54, 93
 - configuration, 59–60
 - Cosmos DB, 55–56
 - data operation, 54
 - timer, 56–58
 - webhooks, 60–63

G

- GetDatabase() method, 195
- GetMetric() method, 211
- GetProperties() method, 97
- GetSubscriber() method, 195
- GetUserDelegationKey() method, 155
- graphs, reviewing, 212–213
- Gremlin API, 68–69

H

- horizontal scaling, 35
- hot partition, 71
- HTTP triggers, 60–63
- hub. *See also* Azure Event Hub; event/s
 - creating, 243–244
 - sending and consuming events from, 244–245

I

- IfNotExists method, 82–83
- image, container, 3
 - building in the cloud, 10–11
 - creating, 3–6
 - publish to the Azure Container Registry, 8–11
- IMDS (Azure Instance Metadata Service), 180
- implementing, change feed notifications, 91–92
- input binding, 48, 50–51
- installation, Docker Desktop, 4
- instance, APIM (Azure API Management) service
 - adding back-end API using OpenAPI specification, 225–226
 - adding back-end API without using a specification, 226–227
 - creating, 224–225
- FROM instruction, Dockerfile, 6

- instrumentation
 - Application Insights, adding, 206–207
 - libraries, 206

J-K

- job, 16, 194
- JWT (JSON web token), 120
- key/s
 - encryption/decryption, 164
 - label attribute, 166
 - naming, 166
 - storing, 177–179
 - value pair, 166, 170–171, 192
- KeyVault API, 172–174, 177–179. *See also* Azure Key Vault
- Kubernetes, 14
- Kudu, 29, 40
- Kusto query language, 214, 216. *See also* query

L

- lease, 103–104
- lease collection, 55
- LFU (Least Frequently Used) algorithm, 193
- library
 - instrumentation, 206
 - Microsoft Authentication, 120
- Library API, applying transformations, 232–233
- life cycle management policy, 107–109
- Log Analytics, 214–215
- log streams, 27
- logical partition, 71
- logs and logging
 - diagnostic, 25–27
 - messages, 212
- loose coupling, 234

M

- managed identity, 180
 - Client ID, 180
 - Principal ID, 180
 - system-assigned, 180–182
 - user-assigned, 180

memory policy

- memory policy, 192–193
- message/messaging, 248
 - Application Insights, 210
 - driven architecture, 234
 - log, 212
 - namespace, 249
 - publishing to the Service Bus Topic, 251–252
 - queuing, 194, 249
 - topic, 249
- metadata, user-defined, 94, 97–98
- method
 - Build(), 170
 - DefaultAzureCredential(), 182–183
 - GetDatabase(), 195
 - GetMetric(), 211
 - GetProperties(), 97
 - GetSubscriber(), 195
 - GetUserDelegationKey(), 155
 - IfNotExists(), 82–83
 - Publish(), 195
 - Release(), 105
 - SetSecretAsync(), 177
 - ToSasQueryParameters(), 156
 - TrackValue(), 211
 - UpsertItemAsync(), 83
 - UseOAuthAuthorizationServer(), 126
- metric/s
 - Application Insights, 208, 210
 - based autoscale rule, 36–38
- Microsoft Entra ID, 141
 - application owner, 144–145
 - managed identity, 180
 - Client ID, 180
 - Principal ID, 180
 - system-assigned, 180–182
 - user-assigned, 180
 - registering your console application, 160–161
 - registering your web application, 142–143
 - web application authentication, 143–144
- Microsoft Graph, 159
 - connectors, 159
 - SDK, 163
 - services, 159
- Microsoft Identity platform, 159
 - components, 120–121
 - endpoint, 120, 121–122
 - management portal, 120
 - OIDC (OpenID Connect), 141
 - wizard, 145

- MongoDB API, 68, 88–89
- monitoring. *See also* Application Insights; Azure Monitor
 - application, 189
 - Azure Function, 241
 - telemetry, 205
- MSAL (Microsoft Authentication Library), 120
- multifactor authentication, 117

N

- namespace, 249–250
- network, content delivery. *See* CDN (Content Delivery Network)
- NoSQL API, 68, 78–82, 85, 88

O

- OAuth, 120
 - authorization server, creating, 124–134
 - basic authentication flow, 123–124
 - creating a client application, 136–139
 - creating a resource server, 134–136
 - roles, 122–123
 - testing the application, 140
- object
 - BlobLeaseClient, 105
 - ConnectionMultiplexer, 194, 198–199
 - DefaultAzureCredentials, 97, 102
- objective mapping, AZ-204 Developing Solutions for Microsoft Azure exam, 263–264
- OIDC (OpenID Connect), 141–142
- OpenAPI, adding back-end API to APIM instance, 225–226
- OpenID, 120
- output binding, 48, 50–52

P

- PaaS (Platform as a Service), 1
- parameter decorators, 50
- parameters, memory policy, 192
- partition
 - Azure Event Hub, 242–243
 - hot, 71

- key, 70
 - choosing, 71–72
 - synthetic, 72
- logical, 71
- physical, 70
- passwordless authentication, 82
- patterns
 - autoscaling, 38
 - binding expression, 51–52
- physical partition, 70
- policy
 - account SAS, 150
 - creating, 231
 - life cycle management, 107–109
 - memory, 192–193
 - retry, 242
 - scope, 231
 - sections, 231
 - Stored Access, 153–154
- PostgreSQL API, 68
- pricing
 - ACR (Azure Container Registry), 9
 - Azure App Configuration store, 165
 - Azure App Service, 25
 - Azure Blob Storage, 106–107
- Principal ID, 180
- processing, event, 246–247
- product, associating with an API, 228
- production environment
 - Dockerfile, 7
 - security, 247
- profile
 - Azure CDN, 200, 202
 - condition, 36–37
- propagation, CDN (Content Delivery Network), 203
- properties, 69
 - Cosmos DB container, 73–74
 - system, 94
- public endpoint, Microsoft Identity platform, 121–122
- Publish() method, 195
- publishing
 - event, 236–238
 - image, 8–11
 - message, 251–252
- pull mode, change feed processor, 90
- push mode, change feed processor, 90

Q

- query
 - Log Analytics, 214–215
 - SQL, 84
- queue, 249
 - creating, 227–256
 - message, 194

R

- reading, change feed notifications, 91
- Redis, 190
 - as a database, 194
 - EXPIRE command, 192
 - as message queue, 195
 - timeout values, 199
- registering your web application in Microsoft Entra ID, 142–143
- Release() method, 105
- replication, Cosmos DB, setting the appropriate consistency level for operations, 86–89
- resource server, creating, 134–136
- resources
 - autoscaling, 34–35
 - metric-based autoscale rule, 36–38
 - patterns, 38
 - profile condition, 36–37
 - rules, 35
 - SAS (shared access signatures), 147
 - account, 148–150
 - token, 148, 158
 - user delegation, 148
 - user delegation, creating, 154–155
- REST API, 224
- retry policy, 242
- revision, 15–16, 229
- roles, OAuth, 122–123
- rules
 - autoscale, 35, 36–38
 - Scale-in/Scale-out, 38
- RUN instruction, Dockerfile, 6
- running a container, 12–14

S

- SAS (shared access signatures), 147
 - account
 - policy, 150
 - URI parameters, 148–149
 - service
 - blob container, 152–153
 - URI parameters, 151–152
 - token, 148, 158
 - user delegation, 148
 - creating, 154–155
 - key, 155–156
- scale rule, 17–18, 35
- Scale-in/Scale-out rule, 38
- scaling
 - Azure Redis Cache, 191
 - container, 2
 - horizontal, 35
 - vertical, 35
- schema
 - agnostic, 73
 - workplace, 214–215
- scope
 - policy, 231
 - subscription, 229
- SDK, 69
 - Azure Cosmos DB, 82
 - Microsoft Graph, 163
 - working with Storage Accounts, 99–106
- secret
 - getting from the key value, 182
 - storing, 177–179
- security. *See also* Microsoft Identity platform
 - authentication, 118
 - API (application programming interface), 229–230
 - client credentials flow, 163
 - form-based, 118
 - Microsoft Entra ID, 143–144
 - multifactor, 117
 - OAuth. *See* OAuth
 - OIDC (OpenID Connect), 141–142
 - token-based, 119–120
 - web application, 121–122
 - authorization, 117
 - connection string, 196
 - production environment, 247
 - selecting, Cosmos DB API, 75
- server
 - authorization, creating, 124–134
 - resource, creating, 134–136
- service, Docker, 7
- service SAS
 - blob container, 152–153
 - URI parameters, 151–152
- service tiers, Azure Redis Cache, 190–191
- SetSecretAsync() method, 177
- setting the appropriate consistency level for operations, Microsoft Identity platform, 121–122
- settings
 - Azure App Service, 165
 - web application, 31–33
 - authentication, 142–143
 - Certificates and Secrets, 143
 - SSL (secure sockets layer), 33–34
- shared key authorization, 147
- sidecar container, 15
- sidecar pattern, 14
- SOAP API, 224
- SQL, query, 84
- SSL (secure sockets layer), web application settings, 33–34
- StackExchange.Redis client, 194
- static content, 189
- static site hosting, Azure Blob Storage, 113–114
- storage, 67
 - authorization, shared key, 147
 - blob, 93
 - access tiers, 106–107, 109–110
 - implement storage policies and data life cycle management, 106–113
 - implementing static site hosting, 113–114
 - interact with data using the appropriate SDK, 99–106
 - lease, 103–104
 - set and retrieve properties and metadata, 94–98
- Cosmos DB, 67–68. *See also* Cosmos DB
 - account, 68
 - APIs, 67–68
 - Cassandra API, 68
 - choosing the partition key, 71–72
 - client authentication, 82
 - Gremlin API, 68–69
 - logical partition, 71
 - MongoDB API, 68
 - NoSQL API, 68

- partitions, 70–71
- physical partition, 70
- PostgreSQL API, 68
- setting the appropriate consistency level for operations, 86–89
- Table API, 68
- partition
 - hot, 71
 - key, 70
 - logical, 71
 - physical, 70
 - synthetic key, 72
- structure, 68
- Stored Access Policy, 153–154
- subscription, 229
 - creating, 229–230
 - event, 235
 - scope, 229
- synthetic partition key, 72
- system properties, 94
- system-assigned managed identity, 180–182

T

- Table API, 68
 - setting the appropriate consistency level for operations, 88
- telemetry, Application Insights, adding, 206–207
- template, webpage, 194
- testing
 - API (application programming interface), 232
 - OAuth, 140
- throughput, Cosmos DB container, 73
- timer triggers, 56–58
- token
 - based authentication, 119–120
 - SAS (shared access signatures), 148, 158
- topic, 235, 249
 - connecting to, 238–240
 - creating, 235–236, 250
 - publishing events to, 236–238
- ToSasQueryParameters() method, 156
- TrackValue() method, 211
- transaction, 194
- trigger
 - configuration, 59–60
 - Cosmos DB, 55–56

- data operation, 54
- function, 47–48, 93
- timer, 56–58
- webhooks, 60–63
- troubleshooting, diagnostic logging, 25–27
- TTL (time-to-live), Azure Redis Cache, 192

U

- updates, AZ-204 Developing Solutions for Microsoft Azure exam, 261–263
- UpsertItemAsync method, 83
- URI parameters
 - account SAS, 148–149
 - service SAS, 151–152
- URL ping test, 216–219
- UseOAuthAuthorizationServer() method, 126
- user delegation SAS, 148
 - creating, 154–155
 - key, 155–156
- user experience, 189
- user session caching, 194
- user-assigned managed identity, 180
- user-defined metadata, 94, 97–98

V

- version
 - API (application programming interface), 229
 - Azure Functions, 56
- vertical scaling, 35
- vertices, 69
- Visual Studio, Forbidden Access Error, 177
- VM (virtual machine), 2, 21
- volume, container, 3

W-X-Y-Z

- web application
 - application owner, 144–145
 - authentication, 22, 121–122, 143–144
 - authorization, 124–134, 146
 - Azure Web App, creating, 200–202
 - caching, 189–190

web application, continued

web application, *continued*

- creating, 121
- deploying code to a, 28–30
- dynamic content, 189
- registering in Microsoft Entra ID, 142–143
- settings, 31–33
 - authentication, 142–143
 - Certificates and Secrets, 143
 - SSL (secure sockets layer), 33–34
 - static content, 189
 - URL ping test, 216–219
- web server, logging, 26–27
- webhooks trigger, 60–63
- webpage template, 194
- wizard, Microsoft Identity platform, 145
- WORKDIR instruction, Dockerfile, 6
- workload profile environment, 15
- workplace schema, 214–215