



# BOOK TITLE: Developing Microsoft Azure Solutions

*Subtitle: The Guide to Cloud Native Applications on Azure*

AUTHOR: Richard Crane

---

## ABOUT THE AUTHOR

In order for us to write your author bio we need a few details. Please remember that your answers should be **relevant** to the book. Your bio helps sell the book so please *only* include *relevant* information:

Full name	Richard Crane
What is your job title?	Founder/CTO
What is the name of the organisation you work for?	MILL5
What is your skillset (as relevant to the book)?	Architect and Developer
In which industry do you work?	Software Consulting
What University degree do you have?	B.S. Electrical and Computer Engineering from Drexel University, Graduated Summa Cum Laude
What professional qualifications do you have?	Microsoft Azure Solutions Architect Microsoft SQL Server Master
Please detail your <i>relevant</i> work history	MILL5, Founder/CTO, 5 Years
	BlueMetal, Chief Architect/CTO, 5 Years
	Grantham, Mayo, Van Otterloo, Architect, 2 Years
	Microsoft, Technical Architect, 10 Years
<i>Relevant</i> projects you have worked on:	
1	Emerson Electric's Plantweb Optics
2	Lionbridge's JobBoard
3	Olympus's MedPresense
4	Fidelity's Active Trader Pro

5	MathWorks' MATLAB Production Server
6	Akamai's Edge Caching for .NET
Would you like to include your social media details (optional)?	Twitter: <a href="#">richcrane</a> LinkedIn: <a href="#">richardcrane</a>

## Can you recommend a good technical reviewer for your book?

James Pansarasa, Chief Architect, MILL5, [jamesp@mill5.com](mailto:jamesp@mill5.com)

Tim Walton, Technical Account Manager, Microsoft, [Tim.Walton@microsoft.com](mailto:Tim.Walton@microsoft.com)

Shri Bhupathi, CoFounder/Technical Fellow, MILL5, [shri@mill5.com](mailto:shri@mill5.com)

Fredrik Estreen, Lionbridge, [Fredrik.Estreen@lionbridge.com](mailto:Fredrik.Estreen@lionbridge.com)

Billy Sells, Emerson, [Billy.Sells@emerson.com](mailto:Billy.Sells@emerson.com)

I have contacts on the Microsoft product teams that we would like to ask to review the book. Once this outline is finished, we would be glad to reach out to those contacts and see if they are willing to review.

# PART ONE: BACKGROUND RESEARCH

## TARGET AUDIENCE

Describe your target reader: what you assume about their knowledge of the topic, related topics, and technical topics generally; why they want to learn the technology; what will they want to do with it?

### *Answer the following:*

Who is your audience?

1	Our audience are professional cloud developers, solution architects, and DevOps engineers interested in building solutions on Microsoft Azure using a cloud-native approach.
2	Prior understanding of Microsoft development tools and frameworks, fundamentals of distributed systems development, and containerization would be necessary.

What is important to them?

1	Those interested in building solutions on Azure need to know many products and services to build good solutions. Learning each of these products and services can take years to fully understand. They need a practical guide to learn these products and services quickly.
2	There are three key challenges: <ul style="list-style-type: none"><li>1) Obtaining the needed skills to create solutions on Azure</li><li>2) Being able to make informed decisions when creating those solutions</li><li>3) Understanding how to manage and monitor solutions on Azure once they are deployed</li></ul>
3	The following are the need to know features: <ul style="list-style-type: none"><li>1) Creating a solution using the various approaches (i.e. products and services)</li><li>2) Scaling solutions in the cloud</li><li>3) Monitoring and maintaining solutions</li></ul>

## COMPETITIVE BOOK TITLES

What is unique about your book? You will need to look on Amazon at books that have been well-received – what are the top three market leading books that your book will compete with? Examine the description, table of contents and book reviews.

**List the books here:**

1	Building Microservices: Designing Fine-Grained Systems, Sam Newman, O'Reilly <a href="#">Click here to go to Amazon</a>
2	Microservices, IoT, and Azure, Bob Fiala, Apress <a href="#">Click here to go to Amazon</a>
3	Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications, Boris Scholl, Trent Swanson & Peter Jausovec, O'Reilly <a href="#">Click here to go to Amazon</a>
4	Exam Ref AZ-203 Developing Solutions for Microsoft Azure <a href="#">Click here to go to Amazon</a>

Please ensure that you have looked at the **description**, **table of contents** and **book reviews** for each of these books.

## OTHER RESOURCES

Developing Microsoft Azure Solutions is meant for readers to pass their Microsoft certifications, specifically AZ-204. We have compiled a list of resources Microsoft provides developers for studying for the AZ-204 exam which will be used to vet the content for this book.

### List the resources here:

1	Microsoft Learning: AZ-204: Developing solutions for Microsoft Azure <a href="#">Click here to go to GitHub (repo)</a>
2	Exam AZ-204: Developing Solutions for Microsoft Azure – Skills Measured <a href="#">Download AZ-204 exam skills outline</a>
3	Exam AZ-203: Developing Solutions for Microsoft Azure – Skills Measured <a href="#">Download AZ-203 exam skills outline</a>
4	Microsoft Learning: AZ-203: Developing solutions for Microsoft Azure <a href="#">Click here to go to Microsoft Learning on GitHub</a>
5	Microsoft Learning: AZ-204: Developing solutions for Microsoft Azure <a href="#">Click here to go to Microsoft Learning on GitHub</a>
6	Microsoft Learning: AZ-900: Microsoft Azure Fundamentals <a href="#">Click here to go to Microsoft Learning on GitHub</a>
7	Microsoft Learning; AZ-300: Microsoft Azure Architect Technologies <a href="#">Click here to go to Microsoft Learning on GitHub</a>
8	Microsoft: Azure Kubernetes Service Workshop <a href="#">Click here to go to Microsoft on GitHub</a>

# PART TWO: BOOK OVERVIEW

---

## OVERVIEW

The long description is the device we use to describe the book on Amazon. Writing it is fairly systematic. Please answer the following questions using only one sentence.

TEMPLATE	EXAMPLE: Mastering Microsoft Power BI	Your turn...
Explain / Introduce the tech	Power BI is a business analytics service that delivers insights to enable fast, informed decisions.	Microsoft Azure is the platform of choice for developing innovative cloud solutions.
Why would a developer want to learn it?	It enables you to go quickly from data to insight that will inform strategic business decisions.	Knowing Azure will help you develop cloud solutions more quickly and easily.
Why should they buy this book?	Mastering Microsoft Power BI provides ultimate coverage of the design, development, management and administration of Power BI content.	Developing Microsoft Azure Solutions will teach you how to build cloud solutions on Azure and will prepare you for taking your Microsoft certification(s) (i.e. AZ-204).
Product approach	Complete with step-by-step explanations of essential concepts, practical examples and self-assessment questions, you will begin by planning a BI project, then examine the development and distribution of content and end with the administration of Power BI for an organization.	Complete with step-by-step explanations of essential concepts, practical examples and self-assessment questions, you will begin with a sample application and examine the development of that application on Azure using App Services, Azure Functions, Container Instances and Azure Kubernetes Service.
Product Breakdown: In 2 sentences, describe the “journey” the book takes the reader on. Look at your section headings for help	You will learn how to create sustainable and impactful datasets, reports, and dashboards. Understand how to connect to data sources, shape and enhance source data, and develop an analytical data model. You will also examine top report and dashboard design practices using features such as Bookmarks and the Power KPI visual.	You will learn how to build modern cloud-native solutions on Azure using container-based applications and microservices. Along the way you will learn the various products and technologies that can be used to build solutions on Azure which will prepare you for taking your Microsoft certifications.
By the end of this book you will...	By the end of this book, you will be confident in creating effective charts, tables, reports or dashboards for any kind of data using the tools and techniques in Microsoft PowerBI.	By the end of this book, you will be able to create a solution on Microsoft Azure using cloud-native techniques such as container-based applications and microservices.
Anything else you would like to add?		Developing Microsoft Azure Solutions is a comprehensive guide to developing solutions on Azure. It covers all the modern techniques for building cloud-native solutions such as serverless computing, container-based applications and microservices.

## LEARNING OUTCOME - WHAT WILL THE READER LEARN AND DO?

Consider the competing books; in particular the **description**, **table of contents** and **book reviews**.

Decide what the key learning objectives will be for your book. List them below:

1	Learn what services and technologies are available in Azure for building solutions
2	Gain an understanding of various Azure products and technologies (ex. App Services and Logic Apps) needed to prepare for your Azure certification(s)
3	Be able to make an informed decision as to which service and technologies to use
4	Build and deploy microservices using Azure Functions, Azure Container Instances (ACI), and Azure Kubernetes Services (AKS)
5	Learn how to scale applications to meet demands and/or reduce costs
6	Best practices for monitoring and maintaining solutions on Azure
7	Learn about modern approaches such as container-based applications and/or microservices for building solutions on Azure

# PART THREE: BOOK STRUCTURE

---

Using your **overview**, and **learning outcomes** now decide on the structure of your book? What are your start and end points?

## GENERAL STRUCTURE

**Divide the book into approximately 3 parts.** The **learning outcomes** you listed previously will help to inform these. These “parts” are a group of chapters that work toward the same goal. Each part will consist of 3-5 chapters. For example: A book on Building Machine Learning Systems with Python might be split into 5 parts as follows: “The Basics”; “Book Learning”; “Numbers, Forecasts and Recommendations”; “Sound and Vision” and finally, “Practical Matters”.

### WRITE YOUR PART HEADINGS BELOW:

1	Building Applications using Azure Compute
2	Developing using Azure Storage and Queues
3	Implementing Security in Azure
4	Monitoring, Troubleshooting and Optimizing
5	Connecting and Integrating Azure Solutions

## CHAPTER OUTLINE

**Each chapter should have a clear focus.** Each chapter title should clearly state what aspect of the overall topic the chapter deals with... Continuing the example of *Building Machine Learning Systems with Python* your section on “Book Learning” might be broken down into 4 chapters as follows: “Clustering – sorting text into groups”, “Topic Modeling – creating non-exclusive groups”; “Logistic Regression – evaluating text quality”; “Bayes Classification – sentiment analysis”. *PLEASE NOTE: Chapter titles appear on Amazon*

<b>PART ONE: Building Applications using Azure Compute</b>	
1	Creating a Web-based application using Visual Studio Code
2	Debugging Locally using Docker and Kubernetes
3	Deploying Web Applications to Azure App Services
4	Deploying Containers to Azure Container Instances (ACI)
5	Creating Serverless Functions with Azure Functions
6	Deploying Applications to Azure Kubernetes Service (AKS)

<b>PART TWO: Developing using Azure Storage and Queues</b>	
7	Developing using Tables, Blobs and Queues
8	Developing using Azure SQL Database
9	Developing using CosmosDB
10	Developing using Azure Service Bus

<b>PART THREE: Implementing Security in Azure</b>	
11	Implementing Authentication and Authorization
12	Managing Keys, Secrets, and Certificates

<b>PART FOUR: Monitoring and Troubleshooting and Optimizing</b>	
13	Monitoring and Troubleshooting in Azure
14	Using Caching to Improve Performance
15	Scaling with Azure Traffic Manager and Front Door



<b>PART FIVE: Connecting and Integrating Azure Solutions</b>	
16	Creating Workflows using Logic Apps
17	Implementing Azure API Management
18	Sending Push Notifications to Users
19	Distributing and Handling Events

**Legend:**

	Topic not included in AZ-204 as of the time of this writing.
	Topic included in AZ-204 as of the time of this writing.

**PLEASE ADD OTHER PARTS/CHAPTERS ONLY IF YOUR TOPIC REQUIRES IT**

# PART FOUR: DETAILED OUTLINE

---

## PART 1: Building Applications using Azure Compute

Our journey will begin by creating an application which can be deployed to Azure. We will start by creating a simple web application using Visual Studio Code, Azure DevOps, and GitHub. This will be the foundation from which we will learn about the various features of Azure.

Once we have the application created, we will examine the many ways which we can deploy the application to Azure. We will start with Azure App Service which is meant for building, deploying, and scaling web applications on Azure. We will then build and deploy our application to containers. Containers are easily preferred way of packaging and deploying modern cloud applications. This will allow us to example container technologies such as Azure Container Instances and Azure Kubernetes Services.

We will then enhance our application with services using Azure Functions. Azure Functions is an event-driven serverless compute platform. Serverless computing is an integral part of cloud platforms to maintain costs while still being able to scale on-demand.

## CHAPTER 1: Creating a Web-based application using Visual Studio Code

- 20-25 pages

### DESCRIPTION:

We will start our journey by creating a simple web-based application which we will expand upon throughout this book. We will setup a project in Visual Studio Code, Azure DevOps and GitHub for the application. The tools used in this exercise are the foundation from which we will build upon.

**Level:** Basic

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Create a web-based application in Visual Studio Code
2. HEADING 2: Store your project in GitHub using Git source control
3. HEADING 3: Setup CI using GitHub Actions
4. HEADING 4: Leverage Azure Boards to manage your work
5. HEADING 5: Knowledge Exam
6. HEADING 6: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Create a simple application using Visual Studio Code
2. SKILL 2: Debug your application locally using Visual Studio Code
3. SKILL 3: Implement CI using Azure Pipelines
4. SKILL 4: Manage epics, user stories, and tasks inside Azure Boards

## CHAPTER 2: Debugging Locally using Docker and Kubernetes

- 20-25 pages

### DESCRIPTION:

Most modern deployments are done using containers which are managed by container orchestrators such as Kubernetes (aka k8s). Container orchestrators automates the deployment, management, scaling, and networking of containers on infrastructure. We will learn how to deploy and debug our application using a container that runs inside Docker and Kubernetes running locally.

**Level:** Basic

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Installing and Running Docker
2. HEADING 2: Setup Kubernetes inside of Docker
3. HEADING 3: Deploying Applications to Containers
4. HEADING 4: Debugging Locally inside Docker and Kubernetes
5. HEADING 5: Knowledge Exam
6. HEADING 6: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to install and configure Docker
2. SKILL 2: Learn how to install and configure Kubernetes running inside of Docker
3. SKILL 3: Learn how to deploy your application to a container
4. SKILL 4: Learn how to debug your application locally inside of Docker and Kubernetes

## CHAPTER 3: Deploying Web Applications to Azure App Service

- 20-25 pages

### DESCRIPTION:

Now that we have our application deployed to a container, we need a way to make our application available to others. We do this by storing our container image inside of a container registry and give access to those that need to deploy our application. We will demonstrate how to create an instance of an Azure Container Registry and publish our application container to the registry.

Once our container is available, we will demonstrate how to deploy our application to Azure App Service. Azure App Service allows developers to build, deploy and scale web applications on Azure. This is the first of several services available in Azure that we can use to deploy and is the simplest for hosting web-based applications on Azure.

**Level:** Basic

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Publishing containers to Azure Container Registry
2. HEADING 2: Creating an App Service
3. HEADING 3: Deploying our application directly to App Service
4. HEADING 4: Deploying our application container to App Service
5. HEADING 5: Scaling using App Service
6. HEADING 6: Knowledge Exam
7. HEADING 7: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create a container registry using Azure Container Registry
2. SKILL 2: Learn how to publish containers to Azure Container Registry
3. SKILL 3: Learn how to create an App Service(s)
4. SKILL 4: Learn how to deploy applications to App Service
5. SKILL 5: Learn how to deploy containers to App Service
6. SKILL 6: Learn how to scale your application using App Service

## CHAPTER 4: Deploying Containers to Azure Container Instances (ACI)

- 8-15 pages

### DESCRIPTION:

We will now examine other approaches for hosting our application in Azure. We will start by leveraging Azure Container Instances (ACIs) which is a service for running serverless containers. ACI allows us to host containers without the need to manage servers. It also provides some of the basic capabilities of a container orchestrator without the need for a container orchestrator like Kubernetes.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Deploy an application to an Azure Container Instance
2. HEADING 2: Container Groups
3. HEADING 3: Persistent Storage
4. HEADING 4: Knowledge Exam
5. HEADING 5: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Deploy an application to Azure Container Instances.
2. SKILL 2: Scale your application based on demand.
3. SKILL 3: Deploy multiple containers to the same container host.
4. SKILL 4: Attach persistent storage to maintain state.

## CHAPTER 5: Creating Microservices using Azure Functions

- 20-25 pages

### DESCRIPTION:

We will now enhance our application to include services based on microservice approach. Microservices is an approach that is often used to simplify the development of your cloud native applications. We will create our services using Azure Functions. Azure Functions allows us to create serverless functions. Serverless functions are small pieces of code which can be hosted and scaled on-demand without the need to manage infrastructure.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Creating a Function
2. HEADING 2: Using Event-based Triggers
3. HEADING 3: Dedicated vs. Consumption Deployment
4. HEADING 4: Creating a Stateful Function using Durable Functions
5. HEADING 5: Knowledge Exam
6. HEADING 6: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create a serverless function.
2. SKILL 2: Invoke the server over different triggers such as HTTP and queues.
3. SKILL 3: Determine a deployment model based on cost and scalability need.
4. SKILL 4: Learn how to create complex stateful processes using Durable Functions.
5. SKILL 5: Identity patterns for creating complex stateful processes.
6. SKILL 6: Know the difference between orchestrator and entity functions.

## CHAPTER 6: Deploying Applications on Azure Kubernetes Service (AKS)

- 20-25 pages

### DESCRIPTION:

We will now learn how to create an Azure Kubernetes Service (AKS) cluster to create and deploy applications on Azure. Kubernetes (aka k8s) is the preferred container orchestrator for deploying, managing and scaling applications containers in Azure. Other popular container orchestrators include RedHat OpenShift and Docker Swarm.

Once deployed, we will debug our application using Azure Dev Spaces. Azure Dev Spaces simplifies the development, deployment, and debugging of applications to Azure Kubernetes Service (AKS) with minimal setup allowing for rapid prototyping and development of your application.

Finally, we will scale Azure Kubernetes Service (AKS) on-demand using virtual nodes and Azure Container Instances (ACIs). This gives us the ability to elastically scale our Kubernetes cluster using Azure.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Create a cluster in Azure Kubernetes Service
2. HEADING 2: Deploy applications to AKS
3. HEADING 3: Deploy and debug an application in an Azure Dev Spaces
4. HEADING 4: Scaling AKS using Azure Container Instances (ACI)
5. HEADING 4: Knowledge Exam
6. HEADING 5: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create a Kubernetes cluster using AKS
2. SKILL 2: Learn how to create Helm Charts which describe the deployment of our applications
3. SKILL 3: Learn how to deploy applications to AKS
4. SKILL 4: Learn how to do Kubernetes development quickly and easily using Azure Dev Spaces
5. SKILL 5: Debug and deploy microservices using Visual Studio Code and Azure Dev Spaces
6. SKILL 6: Leverage routing to debug services using Azure Dev Spaces
7. SKILL 7: Scale AKS on-demand using ACIs
8. SKILL 8: Share state between containers running on AKS and ACI



## PART 2: Developing using Azure Storage and Queues

Almost all applications need a way to persist and maintain state (i.e. data). Azure provides many services including simple services such as Tables and Blobs, message-based services such as Queues and ServiceBus, and complex services such as Azure SQL Database or Azure Cosmos DB.

We will start by examining Tables and Blobs. These services represent fundamental services in Azure to store both structured and unstructured data in Azure. We will start by using these services as the basis for persistence within our application.

Next, we will examine Azure SQL Database and Cosmos DB. These are more complex services for persisting and maintaining state (i.e. data). Azure SQL Database is a cloud-scale relational database engine based on the popular SQL Server Database Engine. Cosmos DB is a globally distributed, multi-model database service. Cosmos DB is much different than a relational database or a simple unstructured store.

Finally, we will examine Queues and ServiceBus which allow for message-based processing. We add functionality to our application to receive and process incoming messages. We will then discuss the various reasons to use messaging such as reliable queuing, publish-subscribe, asynchronous processing, and load balancing.

## CHAPTER 7: Developing using Tables and Blobs

- 15-25 pages

### DESCRIPTION:

Tables and Blobs are fundamental services which represent the basics of storage in Azure. They also represent the basic building blocks for many cloud-native applications. Together we will exam each of these services and where they can be used within our application.

**Level:** Basic

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Create a Storage Account
2. HEADING 2: Storing and retrieving unstructured data using Blobs
3. HEADING 3: Storing and retrieving structured data using Tables
4. HEADING 4: Knowledge Exam
5. HEADING 6: Further Readin

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create a storage account on Azure.
2. SKILL 2: Learn how to store and access unstructured data (ex. files, images, etc.) in Azure
3. SKILL 3: Learn how to store structured data (i.e. relational data) using Tables
4. SKILL 4: Learn how to process incoming messages to perform work using Queues

## CHAPTER 8: Developing using Azure SQL Database

- 20-25 pages

### DESCRIPTION:

Many applications have more complex storage needs. One such need is the ability to store data using a relational database such as Azure SQL Database. Azure SQL Database is based on the popular SQL Server Database Engine. It offers many of the capabilities you would expect from a database such as advanced query processing and high-performance capabilities.

We will start by learning the basics of relational database programming by creating our first database. We will then access that database using an object relational mapper known as Entity Framework Core (aka EF Core). EF Core allows a type-safe way for us to access data stored in databases such as Azure SQL Database.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Create an instance of an Azure SQL Database
2. HEADING 2: Model a simple relational database in Azure SQL Database
3. HEADING 3: Generate a model using Entity Framework Core (EF Core)
4. HEADING 4: Save and query entities within Azure SQL Database
5. HEADING 5: Knowledge Exam
6. HEADING 6: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create an instance of relational database using Azure SQL Database
2. SKILL 2: Learn basic relational data modeling
3. SKILL 3: Use object relational mappers (i.e. Entity Framework Core) to save and access data

## CHAPTER 9: Developing using Azure Cosmos DB

- 20-25 pages

### DESCRIPTION:

Azure Cosmos DB is a globally distributed, multi-model database service from Microsoft running in Azure. It is a multi-model database with support for document, key-value, graph, and column-family data models. We will examine the different APIs supported by Cosmos DB and discuss when we might want to use each one.

**Level:** Advanced

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Selecting an appropriate API for Cosmos DB
2. HEADING 2: Partitioning in Cosmos DB
3. HEADING 3: Using the Table SDK for Cosmos DB
4. HEADING 4: Using the Graph SDK for Cosmos DB
5. HEADING 5: Using the MongoDB API for Cosmos DB
6. HEADING 6: Knowledge Exam
7. HEADING 7: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create an instance of Cosmos DB
2. SKILL 2: Learn what is the appropriate API for Cosmos DB to use within your application
3. SKILL 3: Learn some of the APIs (i.e. Table, Graph, and MongoDB) with Cosmos DB

## CHAPTER 10: Messaging using Queues and ServiceBus

- 15-25 pages

### DESCRIPTION:

Azure supports two types of queues: Azure Queue Storage and Azure ServiceBus. Azure Queue Storage (aka Queues or Storage queues) is based on Azure storage. It provides a reliable way to persist message within and between services using queuing. Azure Service Bus offers queuing capabilities as well, but also offers more advanced messaging capabilities such as in-order processing and publish/subscribe semantics using topics and subscriptions.

We will start by examining Queues to process incoming work based on messages. Messages will be sent allowing us to scale and process messages asynchronously.

Next, we will examine some of the advanced messaging capabilities offered by ServiceBus. We will send messages using publish/subscribe pattern to send messages to one or more subscribers.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Process messages using Queues
2. HEADING 2: Implementing publish/subscribe messaging using ServiceBus
3. HEADING 3: Knowledge Exam
4. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to integrate Azure Storage Queues into applications.
2. SKILL 2: Learn how to integrate Azure ServiceBus into applications.
3. SKILL 3: Know the differences between Queues and ServiceBus

## Olkiijn PART 3: Implementing Security on Azure

There are several features of any application you should consider at the start of development. Security is absolutely one of those features. This includes authenticating and authorizing users as well as protecting the credentials used by our application.

We will start by examining modern techniques for authenticating and authorizing users with Azure Active Directory (AAD). We will then examine external identity providers from Facebook and Microsoft. This will provide us with the basics of authenticating users with our application. We will then use OAuth2 to authorize users in our application.

Next, we will examine how we protected credentials such as keys, secrets, and certificates using Azure Key Vault and Managed Identity. Azure Key Vault provides a programmatic way for developers to securely store and retrieve keys, secrets and certificates. Your code must authenticate with Key Vault to retrieve them requiring credentials in your code. Managed Identity solves this problem

## CHAPTER 11: Implementing Authentication and Authorization

- 20-25 pages

### DESCRIPTION:

Azure Active Directory (AAD) is the way to manage identity information in Azure. It is where most companies manage the identity of their employees. We will examine how we can leverage AAD to authenticate users within our application.

Next, we will examine Azure Active Directory B2C (AAD B2C) to extend identity from your business to consumers. This will allow you to manage identity for customer facing applications. We will then extend that identity to external identity providers such as Facebook and Microsoft.

Finally, we will examine how we can authorize users using OAuth2. OAuth 2.0 is the industry-standard protocol for authorization. We will implement OAuth2 within our application to authorize users.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Authenticate Users using Azure Active Directory (AAD)
2. HEADING 2: Setup Azure B2C to authenticate and authorize users
3. HEADING 3: Connect your application to Facebook
4. HEADING 4: Connect your application to Microsoft
5. HEADING 5: Authorizing using OAuth2
6. HEADING 4: Knowledge Exam
7. HEADING 5: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to integrate Azure Active Directory into your application.
2. SKILL 2: Learn how to authenticate users with external identity providers.
3. SKILL 3: Learn how to connect to Facebook to identity users.
4. SKILL 4: Learn how to connect to Microsoft to identity users.
5. SKILL 5: Learn how to do authorize with OAuth2

## CHAPTER 12: Managing Keys, Secrets, and Certificates

- 15-20 pages

### DESCRIPTION:

Azure Key Vault enables applications and users to store and use credentials that needs to be kept secret. Some common examples include cryptographic keys, passwords, and certificates. We will integrate our application with Azure Key Vault to access credentials needed within our application to access Azure resources.

Finally, we will determine how we can use managed identities to access Azure resources without our application needing to retrieve credentials to access those resources.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Managing Keys, Secrets, and Certificates using Azure Key Vault
2. HEADING 2: Authorize access to resources using Managed Identities
3. HEADING 3: Knowledge Exam
4. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to you Azure Key Vault to store and retrieve credentials.
2. SKILL 2: Learn how to setup managed identities to access Azure resources from your application.
3. SKILL 3: Understand the difference between Azure Key Vault and Managed Identity



## PART 4: Monitoring, Troubleshooting and Optimizing

It is important to integrate monitoring tools into our applications so that we can troubleshoot and diagnose issues that may arise. We will start by integrating Azure Monitor to observe the resources, infrastructure and network of our application. We will then enable Application Insights, a feature of Azure Monitor, to manage the performance of our application. Application Insights is part of a category of tools known as Application Performance Management (APM) which are used to optimize and improve applications.

Lastly, we will examine Azure ARC, a new tool which allows you to extend monitor to any infrastructure, in the cloud, at the edge, on-premises, or in other clouds. Azure ARC has support for many environments including Kubernetes running in your local environment.

### CHAPTER 13: Monitoring and Troubleshooting using Azure - 20-25 pages

#### DESCRIPTION:

We will start by enabling Azure Monitor in our environment and then integrating it into our application. We will use Azure Monitor to monitor our application along with the resources, infrastructure, and network our application is using.

Next, we will enable Application Insights so that we can get an in-depth view of our application. We will examine ways to examine and then improve performance.

Finally, we will use Azure ARC to get a single picture of our application running on premises and in the cloud. We will demonstrate this by monitoring our local Kubernetes cluster.

**Level:** Intermediate

#### Main Chapter Headings (3-5 main chapter headings)

8. HEADING 1: Monitoring using Azure Monitor
9. HEADING 2: Optimizing using Application Insights
10. HEADING 3: Monitoring using Azure ARC
11. HEADING 4: Knowledge Exam
12. HEADING 5: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

6. SKILL 1: Monitor resources, infrastructure and networks for our application using Azure Monitor
7. SKILL 2: Troubleshoot and diagnose performance problems using Application Insights
8. SKILL 3: Learn how to monitor across multiple environments using Azure ARC

## CHAPTER 14: Using Caching to Improve Performance

- 15-18 pages

### DESCRIPTION:

There are times when applications need more performance and scalability than what traditional services have to offer. One commonly used technique to improve performance is caching. Caching offers the ability to store future requests for information so that it can be served faster.

One common way to implement caching is to use an in-memory data store such as Redis. Redis allows one to store data structures in memory forgoing the need to access those data structures from more expensive resources such as databases, files systems, or expensive service requests. We will examine Azure Redis Cache which is an implementation of Redis provided by Microsoft that runs on Azure.

Next, we will use Azure CDN, a content delivery network, to distributed content close to users (i.e. consumers of the content) so that they can access data more quickly. This will improve network bandwidth, latency, and performance of our application.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Implementing caching using Azure Redis Cache
2. HEADING 2: Delivering Content using Azure CDN
3. HEADING 3: Knowledge Exam
4. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to cache expensive operations with Azure Redis Cache.
2. SKILL 2: Deliver content globally using Azure CDN.
3. SKILL 3: Improve application performance by cutting latency and improving performance.

## CHAPTER 15: Scaling with Azure Traffic Manager and Front Door

- 15-18 pages

### DESCRIPTION:

We have discussed numerous ways to scale our application within Azure regions. Now we will examine how to scale our application global using two different approaches: Azure Traffic Manager and Azure Front Door.

Azure Traffic Manager is a global DNS load balancer which allows us to route traffic based on incoming DNS requests. We will examine how to distribute our application across multiple Azure regions and load balance traffic across them.

Azure Front Door is referred to as an Application Delivery Network (ADN). It offers services at the edge to deliver capabilities for your application at the edge. For example, both Traffic Manager and Front Door offer global load balancing, but Azure Front Door also offers TLS termination (aka TLS offloading) at the edge improving performance for your application. We will examine how to load balance your application at that edge using Azure Front Door.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Load balance your application using Azure Traffic Manager
2. HEADING 2: Load balance your application using Azure Front Door
3. HEADING 3: Enable TLS termination with your application using Azure Front Door
4. HEADING 3: Knowledge Exam
5. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Implement global DNS load balancing using Azure Traffic Manager
2. SKILL 2: Implement global load balancing using Azure Front Door
3. SKILL 3: Enable TLS termination using Azure Front Door

## PART 5: Connecting and Integrating Azure Solutions

Many applications need to connect and integrate with other systems and/or users. We will examine several products and technologies in Azure to meet these needs.

We will start our journey by using Logic Apps to build workflows to integrate with external systems. We will examine one of the out-of-the-box connectors for doing this. From there we will use Azure API Management to expose services so that others can integrate with our application.

From there we will examine ways we can connect to users in real-time using Azure SignalR Service and Azure Notification Hubs. We will start by updating information displayed to a user in our application in real-time using web-sockets and SignalR. We will then extend the example to include push notifications sent to mobile users using Notification Hubs.

Finally, we will end our journey by looking at event-based systems using Event Hub and Event Grid. While similar in name, Event Hub and Event Grid are used for two different purposes. We will examine the different use cases so that you know when to use each service.

## CHAPTER 16: Creating Workflows using Logic Apps

- 18-25 pages

### DESCRIPTION:

Azure Logic Apps allows you to build powerful integration solutions. Developers can build business processes and workflows visually that integrate with external systems easily through out-of-the-box connectors (ex. Sales Force, Office 365, Twitter, Dropbox).

Logic Apps supports many other integration needs including business-to-business (B2B) transactions using Electronic Data Exchange (EDI) standards such as EDIFACT, X12, AS2. You can even integrate with your existing on-premise investments in BizTalk Server to extend your business process with cloud-based processes.

We will examine the different ways to create a workflow using Logics Apps using Azure Portal, Visual Studio, and Visual Studio Code. We will then enhance our workflow by using out-of-the-box connectors for integrate with Twitter and send an email.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Create a workflow using Logic Apps in Azure Portal
2. HEADING 2: Create a workflow using Logics Apps in Visual Studio
3. HEADING 3: Create a workflow using Logics Apps in Visual Studio Code
4. HEADING 4: Enhance the workflow with Twitter and Email
5. HEADING 4: Knowledge Exam
6. HEADING 5: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to create workflows using Azure Portal
2. SKILL 2: Learn how to create workflows using Visual Studio
3. SKILL 3: Learn how to create workflows using Visual Studio Code
4. SKILL 4: Learn how to use out-of-the-box connectors in Azure Logic Apps

## CHAPTER 17: Implementing Azure API Management

- 10-15 pages

### DESCRIPTION:

Azure API Management (APIM) is a modern API gateway that allows developers to create consistent and modern APIs to back-end services. It supports advanced features to enhance and extend your existing services. Examples include

We will expose our workflow that we developed using Logic Apps as a service. This service will be protected with authentication using API keys and throttled using denial-of-service prevention. Finally, we will examine the developer portal that we can give to other developers to consume our API.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Expose our workflow using APIM
2. HEADING 2: Protect your APIs using API protection
3. HEADING 3: Hosting a Developer Portal for your APIs
4. HEADING 4: Using APIM to examine users accessing your APIs
5. HEADING 5: Knowledge Exam
6. HEADING 6: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Expose an existing Logic App workflow as an API.
2. SKILL 2: Protect APIs using rate limiting and throttling.
3. SKILL 3: Restrict access to APIs using API keys.
4. SKILL 4: Publish your APIs to other developers using the Developer Portal.

## CHAPTER 18: Sending Push Notifications to Users

- 15-18 pages

### DESCRIPTION:

There are times when you need to notify a user of important information or of an action that needs to be performed. There are many mechanisms to use to accomplish this. Unfortunately, the choice of which mechanism to use depends on how you allow users to interact with your application. For example, if they were using an Apple iPhone, we would need to use Apple Push Notifications (APNS). If they were using an Android Phone, we would need to use Google Firebase Cloud Messaging (FCM). If they were using Windows, we would need to use Windows Push Notifications Services (WPNS).

This could turn into a daunting effort if you need to support multiple platforms in order to interact with your users using notifications. Fortunately, Azure supports Azure Notification Hubs which allows us to communicate with users using a single API across any platform.

While Azure Notification Hubs are great, messages often travel across other networks (ex. Apple or Google) when sending a push notification. That adds latency and security of the push notification. If you need to support real-time messaging and not involve third parties to deliver messages, you can leverage another service within Azure known as Azure SignalR service. This is a real-time communication mechanism using protocols such as web sockets and long polling to send messages directly users.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Sending Push Notifications using Azure Notification Hubs
2. HEADING 2: Communicating with Users using Azure SignalR Service
3. HEADING 3: Knowledge Exam
4. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn the different types of push notifications mechanisms.
2. SKILL 2: Understand the different types of communication protocols for push notifications.
3. SKILL 3: Learn how to implement real-time communication using Azure SignalR Service.
4. SKILL 4: Understand what technology to use for offline notifications.

## CHAPTER 19: Distributing and Handling Events

- 15-25 pages

### DESCRIPTION:

An event is a lightweight notification of a condition or state change. The producer of an event does not have any expectation on how the event is handled. Event-based architectures fit well with cloud-based solutions especially when combined with serverless resources such as Azure Logic Apps and Azure Functions. Azure provides capabilities for building event-based solutions using Event Grid and/or Event Hub.

Azure Event Grid makes it easy to build application using an event-based architecture. Event Grid can process discrete events from event sources, such as blob storage, and process those events by sending them to event handlers (ex. Logic Apps, Azure Functions). Event sources are where the event took place and event handlers are the app or service that reacts to the event.

Azure Event Hubs provides capabilities for processing streams of events and an event ingestion services. Examples include application logging, data archiving, transaction processing, device telemetry streaming (ex. IoT devices), and more.

**Level:** Intermediate

### Main Chapter Headings (3-5 main chapter headings)

1. HEADING 1: Process images saved blob storage using Event Grid
2. HEADING 2: Handle sensor reading from an IoT device using Event Hub
3. HEADING 3: Knowledge Exam
4. HEADING 4: Further Reading

**Skills learned:** *For each heading, insert what the reader will learn to DO in this chapter?*

1. SKILL 1: Learn how to handle events using Event Grid
2. SKILL 2: Learn how to ingest a stream of events using Event Hub