

Mastering Cloud-Native Microservices

*Designing and implementing Cloud-Native
Microservices for Next-Gen Apps*

Chetan Walia



www.bpbonline.com

Copyright © 2023 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-699

www.bpbonline.com

Dedicated to

My mother, who inspired me to pursue my dreams and always supported me at every step of the life. Her love and guidance continue to inspire me. I dedicate this book to her, as a gratitude for her unwavering belief in my potential. Her memory will forever be cherished.

About the Author

Chetan Walia is a senior techno-functional leader and Technology Advisor with over 24 years of experience in Cloud Computing, Digital Transformation, and IT Modernization. He has a proven track record of delivering successful business outcomes through technology innovation, with expertise in Cloud Computing, Digital Transformation, Application Modernization, Telecom Domain, and Financial Planning.

Chetan is an expert in Delivery Management, pre-sales, and client engagement. He has successfully delivered complex Cloud Migration and Digital Transformation programs worth over USD 100 million, and his client portfolio includes Microsoft, AT&T, Amazon Web Services, and more. He is a sought-after speaker, thought leader, and author, with a reputation for grasping the big picture and developing solutions that achieve results.

Chetan has led pre-sales, scoping, and solution-framing sessions and formulated and implemented design principles to strengthen cybersecurity. He has extensive experience in Cloud consulting to build Vision, Strategy, Architecture, and Roadmap.

Chetan's delivery experience includes driving Enterprise Cloud Transformation at Scale, devising and facilitating the biggest Azure Cloud Migration, and establishing Managed Services engagement. He has managed accounts for leading companies in North America, Latin America, and the Asia Pacific regions.

In his personal life, Chetan is a passionate traveler who enjoys exploring the far corners of the world. He is an accomplished photographer with self-published coffee table books based on his own expeditions. He is also passionate about mountains and has experience in high-altitude trekking and mountain-biking.

About the Reviewer

Karthikeyan Shanmugam is an experienced Solutions Architect professional with about 22+ years of experience in the design & development of enterprise applications across industry domains. Currently he is working as Senior Solutions Architect at Amazon Web Services where he helps customers build scalable, secure, resilient and cost-efficient architectures on AWS. Prior to that, he has worked in companies like Ramco Systems, Infosys, Cognizant and HCL Technologies.

He is author of Kubernetes book '**IoT Edge computing with MicroK8s**' published with Packt. His specializations include cloud, cloud-native, containers and container orchestration tools such as Kubernetes, IoT, digital twin and microservices domains and has obtained multiple certifications from various cloud providers.

He is also contributing author in leading journals such as InfoQ, ContainerJournal, DevOps.com, TheNewStack & Cloud Native Computing Foundation (CNCF.io) blog.

His articles on emerging technologies (includes Cloud, Docker, Kubernetes, Microservices, Cloud-native development, etc.) can be read on his blog upnxtblog.com.

Acknowledgement

I would like to express my heartfelt gratitude to the following individuals who have played a significant role in the creation of this book.

First and foremost, I want to thank my father, Sh. Sarabjit Walia, for his firm belief in thinking big and pushing me to reach for the stars. His visionary mindset has been a constant source of inspiration.

I would like to take this opportunity to express my heartfelt gratitude to my sister, Dr. Jyoti, and her husband, Sangram Singh Sandhu. They are avid readers and have extensive collections of books that inspired me to write a book and become part of their book collection.

I am deeply thankful to my brother, Amit Walia, and his wife, Manmeet, for their firm support, encouragement, and motivation at every step of this endeavour. Their belief in my abilities and their constant presence has been invaluable.

Lastly, I would also like to express my appreciation and love to my nephew, Sanmay Walia, a vibrant and energetic four-year-old, whose infectious enthusiasm and boundless energy have kept me motivated. His presence has brought joy and light into my life.

To all those mentioned above and to those who have been there for me, I am truly grateful. Thank you all from the bottom of my heart.

Preface

Microservice architecture is at the heart of cloud-native application architecture, and it has become a crucial tool for companies deploying cloud-based applications. Microservices-based cloud applications are becoming increasingly popular, and enterprises are looking for experienced architects, and DevOps experts who can build, run and develop them. In this book, you will learn how to break down a monolith, create microservices, overcome challenges, and strategize for cloud adoption. '**Mastering Cloud-Native Microservices**' is a guide to help you understand design and implementation steps using industry best practices and design patterns. In a practical case study approach, we will review challenges and solutions faced while identifying and implementing Cloud-native Microservices design patterns.

In this book, readers will learn how to break down a monolithic application into smaller, independent microservices, which can be developed and deployed separately. One of the key benefits of microservices-based cloud applications is that they are designed to take advantage of the elasticity, resiliency, and flexibility of the cloud. The book explores how microservices-based cloud applications can achieve these goals, and provides readers with a comprehensive understanding of the cloud-native concept.

The book is written in an example-driven approach, which makes it easier for readers to understand complex concepts. The book includes case studies that demonstrate how microservices-based cloud applications can be used in real-world scenarios, and provides readers with practical guidance on how to develop and deploy these types of applications.

Key Features:

Comprehensive Coverage: The book covers a wide range of topics related to cloud-native microservices adoption, including modern application design principles, microservice adoption frameworks, design patterns for microservices, cloud-powered microservices, inter-service communication, event-driven data management, the serverless approach, security by design, and cloud migration.

Case study-based approach: The book uses case studies to provide real-world examples of microservices implementation and best practices. This approach helps readers understand how to apply the concepts to their own projects.

Practitioner View: The book provides a practitioner's perspective on cloud-native microservices adoption, making it useful for solution architects, solution experts, pre-sales, and techno-functional roles. It helps readers to understand the challenges and benefits of adopting cloud-native microservices, and how to apply these principles in real-world scenarios.

This preface provides an overview of the chapters you will explore throughout this book, offering a glimpse into the valuable knowledge and insights you will gain.

Chapter 1: Cloud-Native Microservices- In this chapter, we delve into the world of cloud-native microservices, discussing their adoption in modern application architecture. We explore key principles, challenges, and the adoption framework for cloud-native microservices. Five industry success stories demonstrate the transformative power of cloud-native microservices.

Chapter 2: Modern Application Design Principles- The Chapter focuses on the design principles necessary for building resilient, scalable, and performant modern applications. We delve into the Twelve-Factor App methodology and explore design principles for availability, observability, security, and more.

Chapter 3: Microservice Adoption Framework- This chapter provides a structured approach to adopting microservices, covering strategies for breaking down monolithic applications, designing microservices, and building resilient systems. We explore enabling technologies such as Docker and Kubernetes, emphasizing the importance of technology adoption and DevOps processes.

Chapter 4: Design Patterns for Microservices- This chapter delves into essential design patterns for microservices, including integration, database management, observability, and cross-cutting concerns. By understanding and implementing these patterns effectively, you can build scalable and maintainable microservices that meet modern application architecture requirements.

Chapter 5: Cloud-Powered Microservices- In this chapter, we explore the powerful combination of microservices and cloud services. We discuss key design patterns that enhance the capabilities of cloud-powered microservices, such as data management, design and implementation, messaging, and reliability.

Chapter 6: Monolith to Microservices Case Study- The Chapter takes a deep dive into the practical aspects of transitioning from a monolithic architecture to microservices. It explores the challenges faced by legacy systems and provides

effective strategies for updating them. The chapter also covers successful database migration and showcases case studies of practitioners who have implemented microservices.

Chapter 7: Inter-Service Communication- In this chapter, the core concepts are of inter-service communication in microservices architecture. It covers different communication models, including synchronous and asynchronous communication, event-driven communication, and service mesh. The chapter highlights the importance of effective communication patterns for building complex microservices architectures.

Chapter 8: Event-Driven Data Management- The Chapter provides an in-depth discussion of event-driven data management for microservices. It explores technologies like event sourcing and CQRS, event-based data replication, validation, integration, access control, and lineage. The chapter explains how event-driven architectures enable communication between decoupled services and how events can be used to implement business transactions.

Chapter 9: The Serverless Approach- The Chapter explores the serverless approach to microservices development. It covers serverless architecture, frameworks, function-as-a-service platforms, edge computing, monitoring and logging, security, and best practices for serverless microservices development. The chapter showcases case studies of successful serverless microservices implementations.

Chapter 10: Cloud Microservices - Security by Design - The Chapter focuses on building secure microservices through a security-by-design approach. It covers practices for authentication, communication, and data security, container security, monitoring, compliance, infrastructure security, threat detection, and continuous security monitoring. The chapter addresses common security concerns and provides guidance on ensuring the confidentiality, integrity, and availability of microservices-based architectures.

Chapter 11: Cloud Migration Strategy- The Chapter serves as a comprehensive guide to the cloud migration journey. It covers the goals, principles, strategy, and lifecycle stages of cloud migration. The chapter provides an overview of the assessment, planning, design, execution, testing, cutover, and post-cutover stages, highlighting best practices for a successful migration.

I hope that this book will serve as a valuable resource, equipping readers with the knowledge and practical guidance needed to adopt and implement microservices successfully. Each chapter aims to provide in-depth insights, real-world examples, and best practices to ensure a comprehensive understanding of microservices architecture and its related concepts.

Let's embark on this exciting journey into the world of microservices and discover the immense potential it holds for modern application development.

Coloured Images

Please follow the link to download the
Coloured Images of the book:

<https://rebrand.ly/bqagct1>

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At www.bpbonline.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Cloud-Native Microservices.....	1
Introduction.....	1
Structure.....	2
Objectives.....	3
Understanding the cloud native microservices.....	3
Adopting cloud-native microservices	4
Capability maturity level model	7
<i>Focus area: people, process and knowledge to achieve</i>	
<i>End-to-end accountability</i>	<i>9</i>
<i>Focus area: technology and design maturity for enabling</i>	
<i>Zero-touch operations</i>	<i>10</i>
Play book for cloud-native microservices adoption.....	12
Key principles of microservices.....	15
Short case study 01: Snap on AWS.....	18
<i>What can we learn from this example?.....</i>	<i>18</i>
Short case study 02: Wynk Music App.....	19
<i>What can we learn from this example?.....</i>	<i>19</i>
The biggest challenges with microservices adoption.....	20
Short Case Study 03: UPWARD, Inc.	21
<i>What can we learn from this example?.....</i>	<i>22</i>
Short Case Study 04: The Government of India Powers a Population-Scale Vaccine Drive	23
<i>What can we learn from this example?.....</i>	<i>23</i>
SWOT analysis for your application stack.....	24
Short case study 05: IMDb Video Team Builds Strategies for the Future....	26
<i>What can we learn from this example?.....</i>	<i>26</i>
Conclusion.....	28

2. Modern Application Design Principles.....	29
Introduction.....	29
Structure.....	30
Objectives.....	31
Modern application design requirements.....	31
<i>Availability</i>	32
<i>Scalability</i>	33
<i>Performance</i>	34
<i>Observability</i>	34
<i>Security</i>	35
<i>Resiliency</i>	36
<i>Cost optimization</i>	37
<i>Portability, being cloud-agnostic</i>	38
<i>Cloud-native</i>	39
<i>AI/ML enabled</i>	40
<i>DevOps delivery</i>	41
<i>Sustainability</i>	41
The Twelve-Factor App methodology.....	41
<i>Code base</i>	42
<i>Dependencies</i>	43
<i>Configurations</i>	43
<i>Backing services</i>	44
<i>Build, release, run</i>	44
<i>Processes</i>	46
<i>Port binding</i>	46
<i>Concurrency</i>	47
<i>Disposability</i>	48
<i>Dev/Prod parity</i>	49
<i>Logging</i>	49
<i>Admin processes</i>	50
Going beyond the twelve factors.....	50

<i>API first</i>	50
<i>Security</i>	51
Conclusion.....	51
3. Microservice Adoption Framework.....	53
Introduction.....	53
Structure.....	54
Objectives.....	55
From monolith to microservices.....	55
<i>Breaking the monolith: Strategies for building a microservice design</i>	57
<i>Organizing data into bounded contexts or domains</i>	58
<i>Building resilient microservices: Techniques for handling failure and faults</i>	60
<i>Monitoring microservices: Best practices for testing and debugging microservices</i>	61
<i>Embracing continuous delivery with DevOps</i>	62
Enabling technologies for microservices.....	63
<i>Docker and microservices: Use cases for containerization</i>	63
<i>Using Docker: Exploring the benefits of containerization</i>	65
<i>Key components of Docker</i>	66
<i>Container orchestration with Kubernetes</i>	67
<i>Advantages of using Kubernetes: Orchestration for scalability and availability</i>	67
<i>Components of Kubernetes</i>	68
<i>Alternatives to container orchestration: Other tools</i>	70
Microservices adoption using Domain Driven Design.....	71
<i>Domain-driven application decomposition steps</i>	71
Short case study 06: Insurance Claim Processing	72
<i>Using microservices correctly: Characteristics</i>	75
Short case study 07: Modernizes Architecture Using Microservices	77
<i>Learning from the above example</i>	78

<i>Using microservices correctly: Characteristics</i>	78
Conclusion.....	81
4. Design Patterns for Microservices	83
Introduction.....	83
Structure.....	84
Objectives.....	85
Design patterns for microservices.....	86
<i>Decomposition pattern</i>	87
<i>Decompose by business capability</i>	87
<i>Advantages</i>	88
<i>Disadvantages</i>	88
<i>When to use this pattern</i>	88
<i>Decompose by subdomain</i>	89
<i>Advantages</i>	89
<i>Disadvantages</i>	90
<i>When to use this pattern</i>	90
<i>Decompose by transactions</i>	90
<i>Advantages</i>	90
<i>Disadvantages</i>	91
<i>When to use this pattern</i>	91
<i>Decompose by service per team</i>	91
<i>Advantages</i>	92
<i>Disadvantages</i>	92
<i>When to use this pattern</i>	92
<i>Bulkhead pattern for resiliency</i>	93
<i>Advantages</i>	93
<i>Disadvantages</i>	94
<i>When to use this pattern</i>	94
<i>Sidecar pattern for service mesh</i>	94
<i>Advantages</i>	95

Disadvantages.....	95
When to use this pattern.....	96
Strangler pattern for legacy systems	96
Advantages.....	96
Disadvantages.....	97
When to use this pattern.....	97
Integration pattern.....	97
API gateway pattern for API management.....	98
Advantages.....	98
Disadvantages.....	99
When to use this pattern.....	99
API aggregator pattern for composite services.....	99
Advantages.....	100
Disadvantages.....	100
When to use this pattern.....	100
Gateway offloading pattern for performance	101
Advantages.....	101
Disadvantages.....	102
When to use this pattern.....	102
Gateway routing pattern for traffic shaping.....	102
Advantages.....	103
Disadvantages.....	103
When to use this pattern.....	103
Asynchronous messaging pattern for loose coupling	103
Advantages.....	104
Disadvantages.....	105
When to use this pattern.....	105
Branch pattern for parallel processing.....	105
Advantages.....	106
Disadvantages.....	106
When to use this pattern.....	106

<i>Chained microservices pattern for sequencing</i>	106
<i>Advantages</i>	107
<i>Disadvantages</i>	108
<i>When to use this pattern</i>	108
Database management pattern.....	108
<i>Command Query Responsibility Segregator (CQRS)</i> <i>pattern for separation of concerns</i>	108
<i>Advantages</i>	109
<i>Disadvantages</i>	109
<i>When to use this pattern</i>	110
<i>Database per service pattern for decoupling</i>	110
<i>Advantages</i>	111
<i>Disadvantages</i>	111
<i>When to use this pattern</i>	111
<i>Shared database per service pattern for consistency</i>	111
<i>Advantages</i>	112
<i>Disadvantages</i>	112
<i>When to use this pattern</i>	112
<i>Event sourcing pattern for auditing and reconciliation</i>	112
<i>Advantages</i>	113
<i>Disadvantages</i>	114
<i>When to use this pattern</i>	114
<i>Saga pattern for long-running transactions</i>	114
<i>Choreography saga pattern</i>	115
<i>Advantages</i>	116
<i>Disadvantages</i>	116
<i>When to use this pattern</i>	116
<i>Orchestration saga pattern</i>	116
<i>Advantages</i>	117
<i>Disadvantages</i>	117
<i>When to use this pattern</i>	118

Observability pattern.....	118
<i>Distributed tracing pattern for root-cause analysis</i>	118
<i>Advantages</i>	119
<i>Disadvantages</i>	119
<i>When to use this pattern</i>	119
<i>Health check API pattern for self-healing</i>	119
<i>Advantages</i>	120
<i>Disadvantages</i>	121
<i>When to use this pattern</i>	121
<i>Log aggregation pattern for centralized logging</i>	121
<i>Advantages</i>	122
<i>Disadvantages</i>	122
<i>When to use this pattern</i>	122
<i>Application metrics pattern for performance monitoring</i>	123
<i>Advantages</i>	123
<i>Disadvantages</i>	123
<i>When to use this pattern</i>	123
<i>Audit logging pattern for compliance</i>	124
<i>Exception tracking pattern for debugging</i>	124
<i>Monitoring Vs microservices observability</i>	125
Cross-cutting concern pattern.....	126
<i>Blue-green deployment pattern for zero-downtime</i>	126
<i>Advantages</i>	127
<i>Disadvantages</i>	127
<i>When to use this pattern</i>	127
<i>Canary pattern for incremental rollouts</i>	127
<i>Advantages</i>	127
<i>Disadvantages</i>	128
<i>When to use this pattern</i>	129
<i>Canary Vs blue-green deployment pattern for deployment strategies</i>	129
<i>Circuit breaker pattern for fault tolerance</i>	129

<i>Advantages</i>	130
<i>Disadvantages</i>	131
<i>When to use this pattern</i>	131
<i>External configuration pattern for dynamic configuration</i>	131
<i>Service discovery pattern for service registration and discovery</i>	132
<i>Client-side service discovery pattern</i>	133
<i>Server-Side discovery pattern</i>	133
<i>Service discovery methods</i>	134
<i>Advantages</i>	134
<i>Disadvantages</i>	134
<i>When to use this pattern</i>	135
Conclusion.....	135
5. Cloud-Powered Microservices	137
Introduction.....	137
Structure.....	139
Objectives.....	140
Data management design patterns	140
<i>Materialized view</i>	141
<i>Advantages</i>	141
<i>Disadvantages</i>	142
<i>When to use this pattern</i>	143
<i>Sharding</i>	143
<i>Disadvantages</i>	144
<i>When to use this pattern</i>	145
<i>Valet key</i>	145
<i>Advantages</i>	146
<i>Disadvantages</i>	146
<i>When to use this pattern</i>	147
Design and implementation patterns	147
<i>Ambassador</i>	147

<i>Advantages</i>	147
<i>Disadvantages</i>	148
<i>When to use this pattern</i>	148
<i>Anti-corruption layer</i>	149
<i>Advantages</i>	149
<i>Disadvantages</i>	150
<i>When to use this pattern</i>	150
<i>Backends for Frontends</i>	150
<i>Advantages</i>	150
<i>Disadvantages</i>	151
<i>When to use this pattern</i>	151
<i>Leader election</i>	151
<i>Advantages</i>	153
<i>Disadvantages</i>	153
<i>When to use this pattern</i>	153
<i>Messaging design patterns</i>	153
<i>Pipes and filters</i>	153
<i>Advantages</i>	154
<i>Disadvantages</i>	155
<i>When to use this pattern</i>	155
<i>Priority queue</i>	155
<i>Advantages</i>	156
<i>Disadvantages</i>	156
<i>When to use this pattern</i>	156
<i>Publisher-subscriber</i>	156
<i>Advantages</i>	157
<i>Disadvantages</i>	158
<i>When to use this pattern</i>	158
<i>Queue-based load levelling</i>	158
<i>Advantages</i>	158
<i>Disadvantages</i>	159

<i>When to use this pattern</i>	159
<i>Sequential convoy</i>	159
<i>Advantages</i>	159
<i>Disadvantages</i>	160
<i>When to use this pattern</i>	160
Reliability.....	160
<i>Compensating transaction</i>	160
<i>Advantages</i>	161
<i>Disadvantages</i>	161
<i>When to use this pattern</i>	162
<i>Deployment stamps</i>	162
<i>Advantages</i>	163
<i>Disadvantages</i>	163
<i>When to use this pattern</i>	163
<i>Geodes</i>	163
<i>Advantages</i>	163
<i>Disadvantages</i>	164
<i>When to use this pattern</i>	164
<i>Throttling</i>	164
<i>Advantages</i>	165
<i>Disadvantages</i>	165
<i>When to use this pattern</i>	165
Conclusion.....	165
6. Monolith to Microservices Case Study	167
Introduction.....	167
Structure.....	168
Objectives.....	169
Transitioning from monolith to microservices architecture.....	169
<i>Monolithic to microservice design principle</i>	170
Challenges of legacy systems.....	171

Strategies for updating legacy systems to microservices.....	173
Migrating Travelguru application to microservices: A Case Study.....	175
<i>Case Study: Business Challenge</i>	176
<i>Case Study: Solution Delivered for Microservices Migration</i>	177
<i>Target technology stack</i>	179
<i>Case Study: Technology Roadmap for Microservices Adoption</i>	180
<i>Case Study: Application Transition to Microservices Architecture</i>	182
<i>Case Study: Successful Database Migration to Microservices</i>	184
<i>Recommendations to minimize downtime</i>	186
<i>Case Study: Business Outcome of Microservices Migration</i>	187
<i>Case Study: Best Practices Implemented in Microservices Migration</i>	188
Conclusion.....	189
7. Inter-Service Communication.....	191
Introduction.....	191
Structure.....	192
Objectives.....	193
Inter-Service communication.....	193
<i>Challenges of distributed systems</i>	194
<i>Communication models</i>	195
Synchronous inter-service communication.....	196
RESTful APIs	196
<i>Advantages</i>	197
<i>Disadvantages</i>	197
Remote Procedure Calls (RPCs).....	197
<i>Advantages</i>	197
<i>Disadvantages</i>	197
gRPC Remote Procedure Calls	198
<i>Advantages</i>	198
<i>Disadvantages</i>	198
Asynchronous Inter-Service communication	198

<i>Message brokers</i>	199
<i>Advantages</i>	199
<i>Disadvantages</i>	199
<i>Message broker models</i>	200
<i>Message broker software</i>	201
<i>RabbitMQ</i>	201
<i>Advantages</i>	201
<i>Apache Kafka</i>	201
<i>Advantages</i>	202
<i>IBM MQ</i>	202
<i>Advantages</i>	202
<i>Azure service bus</i>	202
<i>Advantages</i>	203
<i>Amazon Simple Queue Service (SQS)</i>	203
<i>Advantages</i>	203
<i>Event-driven communication</i>	203
<i>Publish-subscribe architecture</i>	204
<i>Event-driven architecture</i>	205
<i>Event sourcing</i>	205
<i>Serialization</i>	206
<i>Serialization formats</i>	206
<i>Serialization libraries</i>	206
<i>Best practices for serialization</i>	207
<i>Service mesh</i>	207
<i>Features of service mesh</i>	209
<i>Tools/third-party products for service mesh</i>	211
<i>Istio service mesh</i>	212
<i>Features of Istio</i>	213
<i>Idempotent operations</i>	214
<i>Implementing idempotency</i>	214
<i>Conclusion</i>	215

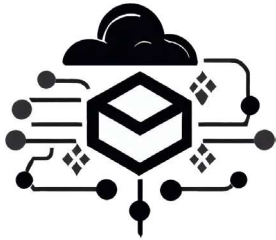
8. Event-Driven Data Management.....	217
Introduction.....	217
Structure.....	218
<i>Objectives</i>	218
Event-driven data management and data governance.....	219
Technologies for event-driven data management.....	221
<i>AWS Kinesis</i>	221
<i>Advantages</i>	222
<i>Disadvantages</i>	222
<i>Google Cloud Pub/Sub</i>	222
<i>Advantages</i>	222
<i>Disadvantages</i>	223
<i>Azure event grid</i>	223
<i>Advantages</i>	223
<i>Disadvantages</i>	224
<i>Apache Kafka on Kubernetes</i>	224
<i>Advantages</i>	224
<i>Disadvantages</i>	224
Event sourcing and CQRS.....	225
Event-based data replication.....	226
<i>Advantages</i>	227
Event-driven data validation.....	228
<i>Advantages</i>	228
Event-driven data integration.....	229
<i>Advantages</i>	229
Event-based data access control.....	230
<i>Advantages</i>	230
Event-based data lineage.....	231
<i>Advantages</i>	231
Data governance in microservices.....	232
Data privacy and compliance.....	233

Data Lifecycle Management.....	234
<i>Advantages</i>	234
Conclusion.....	235
9. The Serverless Approach.....	237
Introduction.....	237
Structure.....	238
Objectives.....	239
Understanding the serverless architecture	239
Use cases for Function-as-a-Service (FaaS).....	240
Serverless framework.....	241
<i>Key features</i>	242
Function-as-a-Service platforms.....	243
<i>AWS Lambda</i>	245
<i>Features of AWS Lambda</i>	245
<i>Advantages of AWS Lambda</i>	246
<i>Disadvantages of AWS Lambda</i>	246
<i>Azure functions</i>	247
<i>Features of Azure functions</i>	247
<i>Disadvantages of AWS Lambda</i>	248
<i>Google cloud functions</i>	248
Serverless approach and edge computing.....	249
Serverless monitoring and logging.....	251
<i>Serverless monitoring and logging is provided by Azure monitor</i>	252
<i>Other popular options</i>	253
Serverless security	253
Best practices for serverless microservices development.....	254
Serverless microservices case studies	256
Conclusion.....	257
10. Cloud Microservices - Security by Design.....	259
Introduction.....	259

Structure.....	261
Objectives.....	261
Cloud Microservices - Security by Design.....	262
Authentication and access control	263
<i>Authentication and authorization mechanisms in cloud microservices</i>	263
<i>Role-based access control (RBAC)</i>	264
<i>Multi-factor authentication (MFA)</i>	265
<i>Access control lists (ACLs)</i>	266
Communication security	267
Data security.....	269
<i>Data security and encryption techniques for microservices</i>	269
<i>Security of data in transit and at rest</i>	270
<i>Immutable infrastructure</i>	271
Container security	272
Monitoring and incident response.....	273
Compliance and risk management	274
<i>Compliance and regulatory considerations</i>	274
<i>Threat modeling</i>	275
<i>Penetration testing</i>	276
Infrastructure security.....	277
Threat detection and response.....	278
Continuous security monitoring.....	279
Conclusion.....	280
11. Cloud Migration Strategy	281
Introduction.....	281
Planning and executing a cloud migration strategy.....	281
Structure.....	282
Objectives.....	284
Cloud migration goals	284
<i>Capex and Opex cost optimization</i>	285

<i>Optimize resource consumption and dynamic elasticity</i>	285
<i>Vendor and application consolidation</i>	286
<i>Agility and innovation via DevOps, Multi-cloud, PaaS, AI/ML, IoT</i>	286
<i>Scalability, flexibility, and global reach</i>	287
<i>Reliability, availability, and security</i>	287
<i>Customer experience and insights</i>	288
<i>IT modernization and integration</i>	288
<i>Reduce, consolidate, and retire the physical data center footprint</i>	289
Cloud migration principles	289
<i>Security first: Secure the network, protect the data, and control access</i>	290
<i>Monitor and optimize workloads for cost</i>	291
<i>Deploy infrastructure as code</i>	291
<i>Make allocations match demand</i>	292
<i>Automate and implement DevOps practices</i>	292
<i>Training the staff for future mode of operations</i>	293
<i>Leverage cloud-native services</i>	293
Cloud migration strategy	293
<i>Business goals and objectives</i>	294
<i>Cloud service provider selection</i>	294
<i>Data security and compliance</i>	295
<i>Cost optimization</i>	295
<i>Scalability and flexibility</i>	295
<i>Legacy systems</i>	295
<i>Change management</i>	296
<i>Performance and reliability</i>	296
<i>Governance</i>	296
<i>Continuous improvement</i>	296
<i>Migration strategy</i>	297
<i>Migration plan</i>	297
<i>Skills and training</i>	297
<i>Performance and optimization</i>	297

<i>Data management</i>	298
<i>Integration</i>	298
<i>Vendor management</i>	298
<i>Stakeholder communication</i>	299
<i>Organizational change management</i>	299
Cloud migration life cycle strategy.....	299
<i>Assessment stage</i>	300
<i>Per application assessment stage</i>	301
<i>Planning stage</i>	303
<i>Design stage</i>	303
<i>Execution stage</i>	303
<i>Testing stage</i>	304
<i>Cutover stage</i>	304
<i>Big Bang Cutover</i>	304
<i>Advantages</i>	304
<i>Disadvantages</i>	305
<i>Phased Cutover</i>	305
<i>Advantages</i>	305
<i>Disadvantages</i>	305
<i>Parallel Cutover</i>	306
<i>Advantages</i>	306
<i>Disadvantages</i>	306
<i>Post cutover stage</i>	306
<i>Optimization stage</i>	307
Conclusion.....	308
Index	309-318



CHAPTER 1

Cloud-Native Microservices

Innovate at Scale with Cloud-Native Microservices

Introduction

Microservices have gained popularity due to their agility and cloud adoption readiness. The combination of microservices and cloud computing has become a key in enterprise architecture, and enterprises continue to reap the benefits of cloud computing. This chapter will cover the industry trends (with short case studies) and how Cloud-native Microservices adoption helping next generation applications. We will discuss the biggest challenges faced by organization when adopting microservices right from culture to technical complexity. Best practices to resolve these challenges and to have a seamless cloud adoption and microservices architecture implementation will be covered throughout this book.

- *According to Technavio the cloud microservices market share is expected to increase by USD 1.59 billion from 2021 to 2026, and the market's growth momentum will accelerate at a CAGR of 25.1%.*
- *According to the Microservices Architecture Market Research Report the Microservices Architecture industry is projected to grow from USD 5.49 Billion in 2022 to USD 21.61 Billion by 2030, exhibiting a compound annual growth rate (CAGR) of 18.66% during the forecast period (2022 – 2030).*
- *According to the 2021 Cloud Native Computing Foundation (CNCF) survey, 92% of respondents reported that they use containers in production, up from 84% in 2020.*

The Cloud Microservices market is growing worldwide, and that is no secret anymore. A large number of companies are upgrading their product portfolios so that they are not only cloud-ready but cloud-native with microservices architecture. The trend is in favour of cloud-native microservices, and organizations that move quickly will have an advantage. There have been tremendous success stories across all industry verticals, including retail and e-commerce, healthcare, telecom/media and entertainment, BFSI, government, and manufacturing. Certainly, there are challenges with respect to investment OPEX/CAPEX, Competitor Landscape, Timelines, ROI for such migrations and modernizations. It is very important for a company to comprehend all the aspects including the risk involved in order to strategize better. Therefore, the key question we should ask yourself would be:

“Cloud-native Microservices” Is it the right choice for your next application?

Companies have shifted from bare-metal infrastructures and monolithic architectures to microservices and container-based architectures in today's fast-paced environment. A better, more flexible, and scalable way to work is being established with cloud native microservices. Organizations can add new application components without revising entire applications or waiting for a release window. In other words, we are able to create new microservices quickly to achieve better results.

In this book, we will examine all aspects of cloud-native microservices, including scalability, flexibility, and resilience. By breaking down applications into small, loosely-coupled services that can be developed, deployed, and scaled independently, cloud-native microservices architecture can help organizations deliver applications faster and with greater reliability. However, adopting cloud-native microservices requires a significant shift in application development, deployment, and management practices, and may not be the best choice for every application. It is imperative to evaluate your specific needs and goals before making a decision. The purpose of this book is to provide you with the right tools, knowledge, and resources to plan, build and implement an optimal roadmap for your business.

Structure

In this chapter we will discuss following topics:

- Understanding the cloud native microservices
- Adopting cloud-native microservices
- Capability maturity level model
 - **Focus area:** People, Process and Knowledge to achieve End-to-end accountability

- **Focus area:** Technology and Design Maturity for enabling Zero-touch operations
- Play Book for cloud-native microservices adoption
- Key principles of microservices
- **Short case study 01:** Snap on AWS
- **Short case study 02:** Wynk Music App
- The biggest challenges with microservices adoption
- **Short Case Study 03:** UPWARD, Inc.
- **Short Case Study 04:** The Government of India Powers a Population-Scale Vaccine Drive
- SWOT analysis for your application stack
- **Short Case Study 05:** IMDb Video Team Builds Strategies for the Future
- Conclusion

Objectives

The purpose of this chapter is to provide you with an overview of the 'Cloud-Native Microservices' adoption framework and why it's the right choice for your next application. To bring the focus back to the three pillars of any transformation, we will discuss the capability maturity level model I devised. We will also talk about how we can move to the next level, whether it's through people, process, or technology. We will discuss the Cloud-native Microservices adoption framework playbook for successful digital transformation in detail. We will be covering five industry success stories to understand the importance of Cloud-Native Microservices and how they can change the game.

Understanding the cloud native microservices

So the key question to start with would be "Is it the right choice for your next application or enterprise?". Rather than being a monolithic entity, cloud-native microservices architectures develop large applications using loosely coupled microservices. A microservice is a small, autonomous, self-contained software component, organized around a business domain, which allows each part to be easily monitored, tested, and updated without affecting the others, enabling greater speed and agility in business and operations.

For any organization to make it a right choice, it is important to develop a strategy with strong governance before moving forward in terms of guiding principles, migration/modernization goals, and business priorities. The technology stack, cloud provider, partners, tools, and so on will be determined by the core strategy and target business value.

In microservices, pieces of code are logically separated so that they can run independently with as few dependencies as possible. However, individual microservices can still communicate with one another and work together to create a complex application. Using self-contained microservices reduces dependencies and the need to coordinate changes across services. As a result, one failed service will have less impact on the entire application.

Containers go well with cloud native microservices and we must plan accordingly when defining a key strategy for such a transformation. The use of containers allows developers to work without constraints imposed by hardware components because they are software-only solutions. Microservices can be designed and tested independently, each performing a specific function.

Likewise, microservices also offer many advantages, but for a successful digital transformation, we need to look beyond technology. It is only through understanding current capability maturity levels and how to reach desired maturity levels encompassing people, process, and technology that we will be able to answer our main question: Is it the right choice for your next application? Let us start with the cloud adoption framework and capability maturity level model.

Adopting cloud-native microservices

Cloud adoption framework required to help clients/organizations build, migrate, modernize, operate, expand, and optimize their applications, infrastructure, data, and analytics in the Cloud. Technology upgrades are only one aspect of the business value equation, but delivering the right value is the key. Hence, we need to consider all aspects like people, process, and technology in our framework.

Next generation application and digital transformation requires industry leaders to rethink their business priorities, strategies, and operations to ensure continued success and value-based delivery. For example, as illustrated in *Figure 1.1*: Key drivers for Cloud-native Microservices adoption can be achieved through a planned adoption.