

Security for Cloud Native Applications

*The practical guide for securing modern
applications using AWS, Azure, and GCP*

Eyal Estrin



www.bpbonline.com

First Edition 2024

Copyright © BPB Publications, India

ISBN: 978-93-55518-903

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

To View Complete
BPB Publications Catalogue
Scan the QR Code:



Dedicated to

The one and only, my beloved wife:

Diana

About the Author

Eyal Estrin is a cloud security architect working with cloud services since 2015. He has been involved in designing and implementing cloud environments from both the IT and security aspects. He has worked with AWS, Azure, and Google Cloud in many different organizations (in the banking, academia, and healthcare sectors). He has attained several top cloud security certifications – CCSP, CCSK, and AWS. He shares his knowledge about cloud security and adoption through social media (LinkedIn, X, Medium, and more) for the benefit of cloud experts worldwide.

About the Reviewer

Israel Chorzevski is a cybersecurity expert and professional white-hat hacker. With wide-ranging experience as VP of Cybersecurity, CTO, Tech-leader, Trainer, Consultant, Red-team, and Software Engineer, Israel ensures that cybersecurity projects are feasible from business and technology perspectives and communicates with all stakeholders in their language.

Over the past 15 years, Israel has gained expertise with multiple technologies, including cloud-native, web, mobile, client-server applications, and IoT/embedded devices. He has worked with various industries, including tech giants, financial institutions, gaming companies, and intelligence agencies.

He dedicates this book to his wife, who consistently encourages him to move forward.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my wife, Diana, who pushed me to achieve my dream of becoming an author.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was an exciting journey revising this book, with the valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues, co-workers, and especially my technical reviewer, who have taught me so much and provided valuable feedback on my work during many years of working in the tech industry.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable

Preface

Cloud is an evolution of IT services, allowing organizations to build highly scalable and secure modern applications.

This book is designed to provide a comprehensive guide to building cloud-native applications. It covers various topics, including using APIs, event-driven architectures, containerization, serverless, CI/CD, and the 12-factor application methodology.

Throughout the book, you will learn about the key features of cloud-native applications and how to use them to build secure modern applications on top of AWS, Azure, and GCP infrastructure.

This book is intended for security professionals, software developers, DevOps, cloud architects, and all those designing, maintaining, and securing cloud-native applications.

This book will give you the knowledge and skills to secure cloud-native applications daily.

Chapter 1: Introduction to Cloud Native Applications - It provides a quick recap of cloud services, an understanding of cloud-native services, and the characteristics of cloud-native applications.

Chapter 2: Securing Modern Design Architectures - It provides a deep understanding of securing modern design architectures from APIs, Event-Driven architectures, and Microservices.

Chapter 3: Containers and Kubernetes for Cloud Native Applications - It provides a deep understanding of securing applications using containers and the Kubernetes platform.

Chapter 4: Serverless for Cloud Native Applications - It explains how Serverless technology can be embedded in cloud-native applications, with recommendations for securing common Serverless technologies from the three major cloud providers.

Chapter 5: Building Secure CI/CD Pipelines - This provides an understanding of the various steps of a CI/CD pipeline and how to embed security controls in each step of the development process.

Chapter 6: The 12-Factor Application Methodology - It provides an understanding of the characteristics of the 12-factor app methodology and how to implement security using containers and Serverless technologies.

Chapter 7: Using Infrastructure as Code - It provides an understanding of IaC for deploying modern infrastructure and explains how to secure common IaC technologies (AWS CloudFormation and HashiCorp Terraform).

Chapter 8: Authorization and Policy as Code - It provides an understanding of Policy as Code and how to implement it as guardrails to enforce settings and implement the authorization decision process.

Chapter 9: Implementing Immutable Infrastructure - It provides a deep understanding of how to implement immutable infrastructure based on the infrastructure of the three major cloud providers.

Chapter 10: Encryption and Secrets Management - It explains how to use key management services and secrets management services to embed encryption and key management as part of cloud-native applications.

Chapter 11: Threat Management in Cloud Native Applications - It provides information on implementing vulnerability management and detecting threats on cloud-native applications at scale.

Chapter 12: Summary and Key Takeaways - It summarizes the topics learned in this book by presenting a sample cloud-native application, how to implement security for this application, and key takeaways from the book.

Code Bundle and Coloured Images

Please follow the link to download the *Code Bundle* and the *Coloured Images* of the book:

<https://rebrand.ly/pjwnkuq>

The code bundle for the book is also hosted on GitHub at

<https://github.com/bpbpublications/Security-for-Cloud-Native-Applications>.

In case there's an update to the code, it will be updated on the existing GitHub repository.

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. Introduction to Cloud Native Applications	1
Introduction	1
Structure	1
Objectives	1
Recap of cloud services	2
Cloud-native services	3
Cloud-native applications.....	4
Conclusion	5
References.....	5
2. Securing Modern Design Architectures	7
Introduction	7
Structure	7
Objectives	7
Application programmable interfaces	8
<i>Understanding APIs</i>	<i>8</i>
<i>Benefits of using APIs</i>	<i>10</i>
<i>Common use cases for using APIs</i>	<i>10</i>
<i>Best practices for securing APIs.....</i>	<i>11</i>
<i>Transport layer</i>	<i>11</i>
<i>Authentication and authorization</i>	<i>11</i>
<i>HTTPS methods.....</i>	<i>12</i>
<i>Input validation</i>	<i>13</i>
<i>API Gateway.....</i>	<i>13</i>
<i>Network and application controls.....</i>	<i>13</i>
<i>Auditing.....</i>	<i>14</i>
<i>Information leakage.....</i>	<i>15</i>
Event-driven architectures.....	15

<i>Understanding Event-driven architecture</i>	15
<i>Pub/Sub model</i>	16
<i>Event streaming model</i>	17
<i>Benefits of using Event-driven architecture</i>	18
<i>Common use cases for using Event-driven architecture</i>	19
<i>External integration</i>	19
<i>Cross-account/Cross-region data replication</i>	19
<i>Business workflow</i>	19
<i>APIs versus Event-driven architecture</i>	20
<i>Communication method</i>	20
<i>Data transfer size</i>	20
<i>Development effort</i>	20
<i>Resiliency to load and failure</i>	20
<i>Best practices for securing Event-driven architecture</i>	21
<i>Network layer</i>	21
<i>Transport layer</i>	22
<i>Encryption at rest</i>	22
<i>Authentication and authorization</i>	23
<i>Auditing</i>	25
<i>Microservices architecture</i>	25
<i>Understanding microservice architecture</i>	26
<i>Benefits of using microservices architecture</i>	26
<i>Decoupled architecture</i>	26
<i>Scalability</i>	27
<i>Fault isolation and resiliency</i>	27
<i>Continuous Integration/Continuous Delivery</i>	28
<i>Language and technology agnostic</i>	28
<i>Common use cases for using microservices architecture</i>	29
<i>Modernizing legacy applications</i>	29
<i>Big data applications</i>	30

<i>Real-time data processing</i>	31
<i>Security in Microservices architecture</i>	31
Conclusion	31
References.....	32
3. Containers and Kubernetes for Cloud Native Applications	33
Introduction	33
Structure	33
Objectives	33
Containers technology.....	34
<i>Understanding Containers</i>	34
<i>Container components</i>	34
<i>Benefits of using containers</i>	36
<i>Excellent use of resources</i>	36
<i>Reduced overhead</i>	36
<i>Small footprint</i>	36
<i>Scalability</i>	36
<i>Portability</i>	36
<i>Speed</i>	36
<i>Developer experience</i>	37
<i>Best practices for securing containers</i>	37
<i>Container registry</i>	37
<i>Least privileged user</i>	37
<i>Read-only file system</i>	37
<i>Container image size</i>	37
<i>Container base image</i>	38
<i>Container image signing</i>	38
<i>Handling third-party vulnerabilities</i>	38
<i>Secrets management</i>	39
<i>Container host</i>	39
<i>Network Layer (Docker images)</i>	39

Container operating systems.....	40
<i>Understanding container operating systems</i>	40
<i>Benefits of container operating system</i>	40
<i>Small footprint</i>	40
<i>Improved security</i>	40
<i>Update mechanism</i>	40
<i>Immutable file system</i>	41
<i>Fast boot time</i>	41
<i>Examples of Container operating systems</i>	41
<i>AWS Bottlerocket</i>	41
<i>Google Container-optimized OS</i>	42
Kubernetes as a Container orchestrator.....	42
<i>Understanding Kubernetes</i>	42
<i>Kubernetes components</i>	43
<i>Control plane</i>	43
<i>Serverless control plane</i>	43
<i>Worker node</i>	44
<i>Benefits of using Kubernetes</i>	45
<i>Run anywhere</i>	45
<i>Automation</i>	45
<i>Community support</i>	45
<i>Cloud support</i>	45
<i>Self-healing capability</i>	45
<i>Horizontal scaling capability</i>	46
<i>Portability and vendor lock-in</i>	46
<i>Cost efficiency</i>	46
<i>Best practices for securing the Kubernetes platform</i>	46
<i>Managed Kubernetes</i>	46
<i>Container OS</i>	47
<i>Confidential computing</i>	47

<i>Pod Security</i>	47
<i>Network layer</i>	48
<i>Pod to Pod communication</i>	49
<i>Service mesh</i>	50
<i>Transport layer</i>	51
<i>Certificate management</i>	51
<i>Encryption at Rest</i>	52
<i>Secrets management</i>	53
<i>Authentication and authorization</i>	53
<i>Configuration standard</i>	55
<i>Security updates</i>	56
<i>Auditing</i>	56
Conclusion	57
References.....	58
4. Serverless for Cloud Native Applications	59
Introduction	59
Structure	59
Objectives	60
Serverless fundamentals	60
<i>Types of Serverless Services</i>	60
<i>Compute</i>	60
<i>Database</i>	61
<i>Storage</i>	61
<i>Application integration</i>	62
<i>Benefits of using Serverless</i>	62
<i>Time to market</i>	62
<i>Scalability</i>	62
<i>High availability</i>	63
<i>Security</i>	63
<i>Cost</i>	63

<i>Introducing Serverless/Function as a Service</i>	63
Introducing AWS Lambda	64
Introducing Azure Functions	65
Introducing Google Cloud Functions	66
<i>Best practices for securing Serverless/Function as a Service</i>	67
<i>Securing Containerized Functions</i>	67
<i>Function isolation</i>	67
<i>Network layer</i>	68
<i>Transport layer</i>	70
<i>Secrets management</i>	70
<i>Authentication and authorization</i>	70
<i>Code signing</i>	71
<i>Vulnerability management</i>	71
<i>Code repository</i>	71
<i>Configuration Management</i>	72
<i>Auditing</i>	73
Conclusion	74
References.....	74
5. Building Secure CI/CD Pipelines	75
Introduction	75
Structure	75
Objectives	76
CI/CD pipeline fundamentals	76
Static Application Security Testing tools	77
<i>Introducing Static Application Security Testing tools</i>	78
<i>Embedding SAST as part of the CI/CD pipeline</i>	78
<i>Examples of open-source SAST tools</i>	78
Software Composition Analysis tools	80
<i>Introducing SCA tools</i>	80
<i>Embedding SCA tools as part of the CI/CD pipeline</i>	80

<i>Examples of open-source SCA tools</i>	80
Static code analyzers for Infrastructure as Code	82
<i>Embedding IaC scanning tools as part of the CI/CD pipeline</i>	82
<i>Examples of open-source IaC scanning tools</i>	82
Repositories and artifacts.....	84
<i>Using repositories as part of the CI/CD process</i>	84
<i>Source code and library repositories</i>	84
<i>AWS CodeCommit</i>	84
<i>Azure Repos</i>	85
<i>Google Cloud Source Repositories</i>	85
<i>Artifact package repositories</i>	85
<i>AWS CodeArtifact</i>	85
<i>Azure Artifacts</i>	86
<i>Google Artifact Registry</i>	86
<i>Container image repositories</i>	86
<i>Amazon Elastic Container Registry</i>	86
<i>Azure Container Registry</i>	87
<i>Google Artifact Registry</i>	87
Software supply chain.....	87
<i>Definition of software supply chain</i>	87
<i>Common threats relating to the software supply chain</i>	87
<i>Introducing Software bill of materials</i>	88
<i>Amazon Inspector</i>	90
<i>Azure SBOM Tool</i>	90
<i>Google Artifact Analysis</i>	90
Best practices for securing the CI/CD pipeline.....	91
<i>Network layer</i>	91
<i>Transport layer</i>	91
<i>Authentication and authorization</i>	91
<i>Design/Plan phase</i>	92

Code development phase.....	92
Build phase.....	92
Test phase.....	92
Delivery phase.....	93
Deployment phase.....	93
Operational/Maintenance phase.....	93
Auditing.....	94
Conclusion.....	94
References.....	94
6. The 12-Factor Application Methodology.....	97
Introduction.....	97
Structure.....	97
Objectives.....	97
The twelve-factor app methodology.....	98
<i>Introduction to the 12-Factors application methodology.....</i>	<i>98</i>
Codebase.....	98
Security best practices.....	99
Dependencies.....	100
Security best practices.....	100
Config.....	101
Security best practices.....	101
Backing services.....	102
Security best practices.....	102
Build, release, run.....	102
Security best practices.....	103
Processes.....	103
Security best practices.....	104
Port binding.....	104
Security best practices.....	105
Concurrency.....	105

<i>Disposability</i>	106
<i>Security best practices</i>	107
<i>Dev/prod parity</i>	107
<i>Security best practices</i>	108
<i>Logs</i>	108
<i>Security best practices</i>	109
<i>Admin processes</i>	109
<i>Security best practices</i>	110
Conclusion	111
References.....	111
7. Using Infrastructure as Code	113
Introduction	113
Structure	113
Objectives	113
Introduction to Infrastructure as Code	114
<i>IaC: Declarative versus imperative</i>	114
<i>Imperative programming</i>	114
<i>Declarative programming</i>	115
<i>Benefits of using IaC</i>	117
AWS CloudFormation	117
<i>Introduction to AWS CloudFormation templates</i>	118
<i>Best practices for securing AWS CloudFormation</i>	119
<i>Identity management</i>	119
<i>Secrets management</i>	119
<i>Parameters management</i>	119
<i>Syntax validation</i>	119
<i>Policy as code</i>	120
<i>Network connectivity</i>	120
<i>Auditing</i>	120
HashiCorp Terraform	120

<i>Benefits of using Terraform</i>	121
<i>Multi-cloud provider support</i>	121
<i>Community support</i>	121
<i>State management</i>	122
<i>Authentication</i>	122
<i>Authorization</i>	122
<i>Best practices for securing Terraform</i>	122
<i>Authentication and authorization</i>	122
<i>Code repository</i>	122
<i>State management</i>	123
<i>Secrets management</i>	123
<i>Static code analysis</i>	124
<i>Policy as Code</i>	124
<i>Auditing</i>	124
<i>CI/CD pipeline</i>	124
<i>Configuration management</i>	124
<i>Using secure Terraform modules</i>	125
<i>Terraform code samples</i>	125
<i>Terraform modules on AWS</i>	125
<i>Terraform modules on Azure</i>	127
<i>Terraform modules on GCP</i>	129
<i>Conclusion</i>	130
<i>References</i>	130
8. Authorization and Policy as Code	131
<i>Introduction</i>	131
<i>Structure</i>	131
<i>Objectives</i>	132
<i>Introduction for Policy as Code</i>	132
<i>Benefits of using Policy as Code</i>	132
<i>Using AWS Service control policies</i>	133

<i>Using Azure Policy</i>	134
<i>Using Google Organization Policy service</i>	135
Introduction to the HashiCorp Sentinel framework	136
<i>Using Sentinel to complement Terraform modules</i>	136
<i>Code samples for Sentinel policies</i>	137
Introduction to Open Policy Agent	139
<i>Benefits of using OPA</i>	139
<i>Authorization process using OPA</i>	140
<i>Sample “Hello World” policy</i>	141
<i>Sample code for using OPA to secure Kubernetes</i>	142
Introduction to Cedar policy language	144
<i>Authorization process using Cedar</i>	144
<i>Sample Cedar code</i>	145
Conclusion	148
References	149
9. Implementing Immutable Infrastructure	151
Introduction	151
Structure	151
Objectives	151
Introduction to immutable infrastructure	152
<i>Differences between stateful and stateless applications</i>	152
<i>Introducing Immutable Infrastructure</i>	153
<i>Benefits of using immutable infrastructure</i>	153
Building a golden image	154
<i>Best practices for creating container golden image</i>	154
<i>Virtual machine image source</i>	154
<i>Virtual Machine Image update</i>	155
<i>Virtual Machine Image builder</i>	155
<i>Container Image source</i>	156
<i>Container Image Builder</i>	156
<i>Container registry</i>	157

<i>Managing persistent data</i>	157
<i>Managing environment variables</i>	157
<i>Secrets management</i>	157
Creating deployment pipeline	158
<i>Implementing Immutable Infrastructure as part of the CI/CD pipeline</i>	158
<i>CI/CD pipeline using AWS services</i>	158
<i>CI/CD pipeline using Azure services</i>	159
<i>CI/CD pipeline using GCP services</i>	160
<i>CI/CD pipeline using vendor-agnostic tools</i>	161
Conclusion	162
References	162
10. Encryption and Secrets Management	163
Introduction	163
Structure	163
Objectives	163
Introducing encryption and key management services	164
<i>Introducing key management services</i>	165
<i>Best practices for securing key management services</i>	166
<i>Introduction to AWS KMS</i>	166
<i>Best practices for securing AWS KMS</i>	168
<i>Introduction to Azure Key Vault</i>	168
<i>Best practices for securing Azure Key Vault</i>	169
<i>Introduction to Google Cloud KMS</i>	170
<i>Best practices for securing Google Cloud KMS</i>	171
Introduction to secrets management in cloud-native applications	172
<i>Secrets management risks</i>	172
<i>Best practices for securing secrets management services</i>	173
<i>Introduction to AWS Secrets Manager</i>	173
<i>Best practices for securing AWS Secrets Manager</i>	174
<i>Secrets Management in Azure</i>	175

<i>Best practices for securing secrets using Azure Key Vault</i>	175
<i>Introduction to Google Secret Manager</i>	175
<i>Best practices for securing secrets using Google Secret Manager</i>	176
<i>Introduction to HashiCorp Vault</i>	176
<i>Best practices for securing secrets using HashiCorp Vault</i>	177
<i>Secrets management in Git repositories</i>	178
<i>Secrets management in the CI/CD pipeline</i>	179
<i>AWS CodeBuild</i>	179
<i>Azure DevOps pipelines</i>	179
<i>Google Cloud Build</i>	180
<i>Secrets management in Containers</i>	180
<i>Scanning for secrets inside Container images</i>	180
<i>Securing access to secrets in Kubernetes</i>	181
<i>Secrets management in Function-as-a-Service</i>	183
<i>AWS Lambda</i>	183
<i>Azure Functions</i>	183
<i>Google Cloud Functions</i>	184
<i>Secrets management in Infrastructure-as-Code</i>	184
<i>Conclusion</i>	185
<i>References</i>	185
11. Threat Management in Cloud Native Applications	187
<i>Introduction</i>	187
<i>Structure</i>	187
<i>Objectives</i>	187
<i>Vulnerability versus threat versus risk</i>	188
<i>Introducing vulnerability management in Cloud-native applications</i>	188
<i>Introduction to Amazon Inspector</i>	189
<i>Amazon Inspector for Containers</i>	189
<i>Amazon Inspector for Lambda</i>	190
<i>Best practices for implementing Amazon Inspector</i>	191

<i>Introduction to Microsoft Defender for Cloud</i>	191
<i>Microsoft Defender for Containers</i>	191
<i>Microsoft Defender for Cloud DevOps Security</i>	193
<i>Best practices for implementing Microsoft Defender for Cloud</i>	193
<i>Introducing GitHub advanced security for Azure DevOps</i>	194
<i>Best practices for implementing GitHub Advanced Security for Azure DevOps</i>	194
<i>Introducing Google vulnerability management services</i>	195
<i>Best practices for implementing Google vulnerability management services</i>	197
Implementing threat intelligence at scale	197
<i>Introduction to Amazon GuardDuty</i>	197
<i>Best practices for implementing Amazon GuardDuty</i>	200
<i>Introducing Microsoft Sentinel</i>	200
<i>Best practices for implementing Microsoft Sentinel</i>	201
<i>Introducing Google Security Command Center</i>	202
<i>Best practices for implementing Google Security Command Center</i>	203
Conclusion	204
References.....	204
12. Summary and Key Takeaways	205
Introduction	205
Structure	205
Objectives	205
Introducing Pet Store.....	206
Key takeaways from the book.....	207
<i>Chapter 1, Introduction to Cloud Native Applications: Key takeaways</i>	208
<i>Chapter 2, Securing Modern Design Architectures: Key takeaways</i>	208
<i>Chapter 3, Containers and Kubernetes for Cloud Native Applications: Key takeaways</i>	209
<i>Chapter 4, Serverless for Cloud Native Applications: Key takeaways</i>	210
<i>Chapter 5, Building Secure CI/CD Pipelines: Key takeaways</i>	210
<i>Chapter 6, The 12-Factor Application Methodology: Key takeaways</i>	210
<i>Chapter 7, Using Infrastructure as Code: Key takeaways</i>	211

<i>Chapter 8, Authorization and Policy as Code: Key takeaways</i>	211
<i>Chapter 9, Implementing Immutable Infrastructure: Key takeaways</i>	212
<i>Chapter 10, Encryption and Secrets Management: Key takeaways</i>	212
<i>Chapter 11, Threat Management in Cloud Native Applications: Key takeaways</i>	213
Recommendations for the readers of the book	213
<i>Gain hands-on experience</i>	213
<i>Share knowledge with your peers</i>	214
<i>Learn from experts</i>	214
Index	215-224

CHAPTER 1

Introduction to Cloud Native Applications

Introduction

This chapter will provide you with a quick recap of cloud services, an understanding of cloud-native services, and the characteristics of cloud-native applications.

Note: In various chapters of the book, we will mention services from AWS, Azure, and GCP and provide best practices for securing those services. The services will be ordered alphabetically according to the cloud providers' names.

Structure

The chapter covers the following topics:

- Recap of cloud services
- Cloud-native services
- Cloud-native applications

Objectives

At the end of this chapter, you will be able to understand the different cloud service models and cloud-native services, and you will be able to recognize cloud-native applications.

Recap of cloud services

Before we dive into cloud-native applications, let us have a short recap of what the cloud service models and how they relate to cloud-native applications.

The **National Institute of Standards and Technology (NIST)** provides us with the following definitions:

- **Infrastructure as a Service (IaaS)**

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, which can include operating systems and applications.

- **Platform as a Service (PaaS)**

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

- **Software as a Service (SaaS)**

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (for example, web-based email), or a program interface.

(Source: NIST Special Publication 800-145: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>)

The shared responsibility model is a concept that tries to set the boundaries of responsibility between the **cloud service provider (CSP)** and the cloud consumer (or customer) in various topics such as security, availability, and even sustainability. The CSP is responsible for the infrastructure layers from physical data centers to computing, storage, network, and virtualization. When customers choose IaaS, they are responsible for everything within operating systems, such as runtime and application configuration. When customers choose PaaS, they may have the option to select the runtime version and import their code into a managed service environment, depending on the service offered by the CSP. When customers choose SaaS, they only control their data. In all service models, the customers are always responsible for deciding which data to store in the cloud and who has access to their data.

Refer to *Figure 1.1*: