

Cloud Native Microservices Cookbook

*Master the art of microservices in the
cloud with over 100 practical recipes*

Varun Yadav



www.bpbonline.com

First Edition 2024

Copyright © BPB Publications, India

ISBN: 978-93-55519-603

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:



www.bpbonline.com

Kup książkę

Dedicated to

My beloved parents:

Krishan Kumar Yadav

and

Manju Yadav

Your nurturing guidance and unwavering support have shaped me into the person I am today. With your love and wisdom, you have empowered me to confidently navigate life's journey and make conscientious decisions. Thank you for providing me with a foundation to stand tall and pursue my dreams.

And to my loving wife:

Neha Yadav

Your enduring companionship and intense faith in my abilities have served as my steadfast source of strength and inspiration. I express my deepest gratitude for being my pillar of support and my greatest source of joy.

About the Author

Varun Yadav, a seasoned Computer Science Engineer with nearly two decades of rich expertise spanning Enterprise Architecture, Solution Architecture, Cloud Computing, Microservices, and DevOps. Combining technical acumen with strategic vision, he holds an MBA in Information Management and Strategy, complemented by a Post Graduate Diploma in AI and Machine Learning. With a career deeply rooted in the BFSI domain, Varun has been a driving force behind innovation, actively participating in Proof of Concepts (POCs), leading Idea Generation initiatives, and igniting creativity in Hackathons. Notably, Varun believes in leading by example, having successfully helmed large-scale teams through intricate projects and endeavors. With an unwavering commitment to staying ahead of the curve in technology, he embodies a blend of expertise and foresight, consistently delivering transformative solutions that redefine industry standards. Whether forging innovative approaches or fostering collaborative environments, he stands as a beacon of excellence, guiding teams toward unparalleled success in the ever-evolving landscape of technology.

About the Reviewers

- ❖ **Kumar Gaurav**, with over 12 years of experience in software engineering, is currently serving as an Architect at LTI-Mindtree, where he leads digital transformation initiatives. Specializing in open-source solutions and cloud-native technologies, he ensures the implementation of industry best practices in infrastructure setup, development, deployment, and monitoring for large enterprises. Kumar's expertise encompasses Kubernetes, OpenShift, Docker, Git, Java, Microservices, API Gateway, ESB, WSO2, DevOps MySQL, Linux, tool based development and more.

Certified in WSO2 & Oracle middleware technologies, Kumar excels in on-premise and cloud-based implementations, with proficiency in Amazon Web Services, Google Cloud, and Oracle Cloud. He has played a pivotal role in end-to-end project implementation and post-implementation support across various domains such as Finance, Railways, Healthcare, Power sector, and Business Intelligence.

Driven by a passion for innovation and fostering seamless communication between systems, Kumar actively seeks networking and collaboration opportunities within the realm of modern digital architecture.

- ❖ **Sumit Bajaj**, a seasoned Engineering Leader and Integration Solution Architect with 20 years of rich experience in designing and developing software products to solve complex business problems. A creative problem solver and solution-oriented professional, dedicated to improving business operations and meeting client technical needs. With proven leadership characteristics, he drives enterprise-grade solutions from conception to completion, consistently delivering exceptional project outcomes.

Acknowledgement

I am deeply indebted to my wife, Neha, whose constant backing made it possible for me to dedicate countless hours to the creation of this book. Thank you for your patience, and encouragement, and for being my pillar of strength throughout this journey. Your love and belief in me mean everything.

I extend my sincere gratitude to all my managers who have been more than just colleagues; they have been mentors, guides, and friends. Special thanks to Marul Mehta for his invaluable support and confidence in me during the formative stages of my career, and to Rakesh Mahajan for his encouragement to embark on this writing journey. Your belief in my abilities and encouragement has been instrumental in making this book a reality.

I would also like to express my heartfelt appreciation to BPB Publications for giving me the opportunity to write this book. It has been an honor to collaborate with such a reputable publishing house, and I am grateful for the trust they have placed in me.

Preface

Welcome to the preface of this book, where I am thrilled to share my journey and passion for technology with you. From the very beginning, I have been captivated by the wonders of technology, finding immense joy in seeing my contributions come to life and being used by people. It has been an exhilarating ride dedicating my entire career to crafting and delivering technology solutions, particularly in the BFSI domain.

What drove me to write this book? Well, it is simple. I have always craved the full picture – the complete lifecycle, the entire technology stack. That burning curiosity inspired me to embark on this writing journey. This book is not just a collection of recipes; it is a meticulously planned expedition through the intricate world of designing and architecting microservices from scratch, all the way to harnessing the boundless potential of the cloud.

Think of it as your guide to launching microservices in the cloud – a comprehensive roadmap that covers every aspect, from the nitty-gritty of developing microservices to the intricacies of hosting and administrating them in the cloud. Through real-world insights and practical examples, I aim to demystify the complexities of proficient system architectures, equipping you with the tools and know-how to navigate this exciting terrain with confidence.

Chapter breakdown:

Chapter 1: Microservices and Cloud - Discover the foundational concepts of microservice architecture and cloud elasticity. Delve into the importance of Java frameworks and learn to configure cloud services and setup cloud infra for optimal performance.

Chapter 2: Developing Microservices and Test Cases - Establish a solid foundation for microservice development, starting with environment setup and the API-first approach. Explore practical examples and running code snippets to kickstart your journey.

Chapter 3: Externalize Application Configurations - Learn to externalize application configurations for enhanced flexibility and seamless deployment across different environments. Dive into Spring Cloud Config Server and Spring Actuator for efficient configuration management.

Chapter 4: Implementing Dynamic Services - Explore the need for service discovery and various approaches to achieve seamless communication between dynamically changing microservices.

Chapter 5: Containerization Using Docker - Unlock the agility and efficiency of containerization with Docker. Master the process of wrapping applications in lightweight images and deploying them to AWS ECR.

Chapter 6: Pipeline Automation for CI/CD - Streamline your development process with automated CI/CD pipelines. Discover best practices for code quality checks, test case execution, and security assessments.

Chapter 7: Microservices Orchestration - Centralize control and management of containerized applications with microservices orchestration. Learn to set up orchestrators, fetch from registries, and scale your applications seamlessly.

Chapter 8: Auto Scalability, High Availability, and Disaster Recovery - Ensure high availability and scalability of your applications in the AWS cloud. Explore strategies for disaster recovery and handle scale-out and scale-in events effectively.

Chapter 9: Cloud Security - Prioritize security considerations for your microservices in the cloud. Discover techniques to safeguard your applications using available cloud security offerings.

Chapter 10: Observability - Gain insights into the internal state of your applications with observability. Learn to set up logs, metrics, and tracing for better understanding and critical observations about the application behavior.

Chapter 11: Infrastructure Automation with IaC - Simplify infrastructure setup with Infrastructure as Code (IaC). Explore automation techniques for creating and managing infrastructure across different environments.

So, buckle up, dear reader, as we embark on this thrilling adventure together. Let us dive deep into the world of microservices and cloud technologies, and uncover the secrets to building robust, scalable solutions that stand the test of time.

Happy reading and hands-on exploring!

Code Bundle and Coloured Images

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

<https://rebrand.ly/gah3typ>

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

<https://github.com/bpbpublications/Cloud-Native-Microservices-Cookbook>.

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. Microservices and Cloud	1
Introduction	1
Structure	1
Objectives	2
Importance of microservices	2
<i>Development</i>	3
<i>Testing</i>	3
<i>Deployment</i>	3
<i>Management</i>	4
Importance of cloud	5
Using the Java programming language	6
Java frameworks for microservices	8
Configuring cloud services	9
<i>Recipe 1: Installing AWS command line interface</i>	9
<i>Installing AWS CLI on Windows</i>	9
<i>Installing AWS CLI on Linux</i>	10
<i>Installing AWS CLI on MacOS</i>	11
<i>Recipe 2: Creating IAM admin user</i>	12
<i>Steps to create an admin user</i>	12
<i>Recipe 3: Configuring IAM admin user in the AWS CLI</i>	13
<i>Recipe 4: Creating an S3 bucket</i>	13
<i>Creating an S3 bucket using AWS Management Console</i>	13
<i>Creating an S3 bucket using AWS CLI</i>	14
<i>Recipe 5: Creating Postgres DB</i>	15
<i>Creating a PostgreSQL database using AWS Management Console</i>	15
<i>Creating a PostgreSQL database using AWS CLI</i>	16
Conclusion	19
Points to remember	19
Questions	19

2. Developing Microservices and Test Cases	21
Introduction	21
Structure	21
Objectives	22
Application description.....	22
Development environment setup	22
<i>Recipe 6: Creating Maven projects.....</i>	<i>22</i>
Customer Enquiry	23
Enquiry Broadcast	23
<i>Recipe 7: Enhance pom.xml files</i>	<i>24</i>
Customer Enquiry	24
Enquiry Broadcast	26
<i>Recipe 8: Creating Spring Boot application class</i>	<i>28</i>
Customer Enquiry	28
Enquiry Broadcast	29
Running applications.....	29
<i>Recipe 9: Connecting RDS PostgreSQL instance.....</i>	<i>31</i>
<i>Recipe 10: Creating database schemas.....</i>	<i>32</i>
Customer Enquiry	32
Enquiry Broadcast	33
Version control systems.....	33
<i>Recipe 11: Creating and configuring Git repositories</i>	<i>34</i>
Customer Enquiry	35
Enquiry Broadcast	36
Create merge request	37
API-first approach.....	38
Advantages of API-first approach	38
Developing microservices.....	39
<i>Recipe 12: Creating database tables.....</i>	<i>39</i>
Customer Enquiry	40

<i>Enquiry Broadcast</i>	42
<i>Recipe 13: Developing Customer Enquiry service</i>	42
<i>Adding dependency</i>	43
<i>Adding application.properties</i>	43
<i>Creating RESTController</i>	44
<i>Defining financial products</i>	46
<i>Creating service classes</i>	47
<i>Creating entity classes</i>	50
<i>Creating repositories</i>	55
<i>Code compilation</i>	56
<i>Recipe 14: Developing Enquiry Broadcast service</i>	57
<i>Adding dependency</i>	57
<i>Appending application.properties</i>	58
<i>Creating RESTController</i>	58
<i>Creating config class</i>	59
<i>Creating service class</i>	60
<i>Creating entity class</i>	64
<i>Creating repositories</i>	66
<i>Code compilation</i>	66
<i>Recipe 15: Creating unit test cases</i>	67
<i>Adding dependency</i>	67
<i>Writing test classes</i>	67
<i>Recipe 16: Enhancing microservices for interservice communication</i>	72
<i>Adding dependency</i>	72
<i>Service changes</i>	73
<i>Recipe 17: Running the application</i>	75
<i>Recipe 18: Pushing the code to Git repositories</i>	77
<i>Conclusion</i>	77
<i>Points to remember</i>	77
<i>Questions</i>	78

3. Externalize Application Configurations	79
Introduction	79
Structure	79
Objectives	80
Application configuration.....	80
@Value annotation	80
Recipe 19: Using @Value to inject value as a parameter	81
Adding a property	81
Using the property	81
Profile configuration files.....	82
Recipe 20: Using profile specific properties	82
Creating application-dev.properties	82
Activating profile	82
Externalizing configurations	83
External configuration files	83
Recipe 21: Using external configuration file.....	83
Parameterizing location of Broadcast Enquiry microservice	84
Externalizing Broadcast Enquiry location.....	84
Environment variables	85
Recipe 22: Using environment variables.....	85
Command line arguments	85
Recipe 23: Using command line arguments.....	85
Configuration override order.....	86
Version controlled configurations.....	86
Recipe 24: Creating Spring Cloud Config Server	86
Configure server Maven project.....	87
Creating server application class	89
Creating configuration Git repository	90
Point Config Server to configuration Git Repository	91
Accessing configurations	92
Config server details in microservices.....	94

Secured configurations	96
<i>Recipe 25: Integrating with HashiCorp Vault</i>	96
<i>Setting up a Vault</i>	96
<i>Storing secrets in a Vault</i>	98
<i>Config Server with composite backend</i>	100
Dynamic configurations	102
<i>Recipe 26: Updating configurations at runtime</i>	102
<i>Configuring Spring Actuator</i>	102
<i>Using @RefreshScope</i>	103
<i>Validating the solution</i>	104
Conclusion	105
Points to remember	106
Questions	106
4. Implementing Dynamic Services	107
Introduction	107
Structure	107
Objectives	108
Importance of service discovery	108
<i>Recipe 27: Creating discovery server</i>	109
<i>Configure server Maven project</i>	110
<i>Creating server application class</i>	112
<i>Configuring discovery server</i>	113
Service discovery patterns	114
<i>Client-side service discovery</i>	114
<i>Recipe 28: Client-side discovery using Eureka server</i>	115
<i>Registering service with Eureka server</i>	115
<i>Service discovery and invocation</i>	117
<i>Server-side service discovery</i>	118
<i>Recipe 29: Creating gateway server</i>	119
<i>Configure server Maven project</i>	120

Create server application class	122
Configuring routing configuration	122
Configuring gateway server	123
Recipe 30: Server-side discovery using Spring Cloud Gateway	124
Configuring gateway server URL	124
Gateway server invocation and service discovery	124
Client-side versus server-side service discovery	126
Service registration patterns	127
Self-registration	127
Third-party registration	128
Conclusion	129
Points to remember	129
Questions	130
5. Containerization Using Docker	131
Introduction	131
Structure	131
Objectives	132
Introduction to containers	132
Open Container Initiative	133
Docker	135
Installing Docker	136
Recipe 31: Installing Linux Docker Engine on Linux	137
Uninstalling old versions	137
Setting up the apt repository	137
Installing Docker Engine	140
Recipe 32: Installing Docker Desktop on Windows	142
Building Docker images	143
Dockerfile	143
.dockerignore file	145
Third-party registration pattern	146
Entry point script	147

<i>Recipe 33: Creating entry point scripts.....</i>	<i>147</i>
<i>Customer enquiry entry point script</i>	<i>147</i>
<i>Enquiry Broadcast entry point script</i>	<i>149</i>
<i>Config Server entry point script</i>	<i>150</i>
<i>Recipe 34: Creating .dockerignore.....</i>	<i>151</i>
<i>Customer Enquiry .dockerignore</i>	<i>151</i>
<i>Enquiry Broadcast .dockerignore</i>	<i>152</i>
<i>Enquiry Broadcast .dockerignore</i>	<i>152</i>
<i>Recipe 35: Creating Dockerfile</i>	<i>153</i>
<i>Customer Enquiry Dockerfile</i>	<i>153</i>
<i>Enquiry Broadcast Dockerfile</i>	<i>155</i>
<i>Config Server Dockerfile</i>	<i>156</i>
<i>Recipe 36: Build Docker image using CLI</i>	<i>157</i>
<i>Building Customer Enquiry Docker image</i>	<i>157</i>
<i>Building Enquiry Broadcast Docker image</i>	<i>158</i>
<i>Building Config Server Docker image</i>	<i>158</i>
<i>Verifying created Docker images.....</i>	<i>158</i>
<i>Running Docker images.....</i>	<i>158</i>
<i>Recipe 37: Run Docker image using CLI</i>	<i>159</i>
<i>Running Config Server Docker image.....</i>	<i>160</i>
<i>Running Customer Enquiry Docker image.....</i>	<i>160</i>
<i>Running Enquiry Broadcast Docker image.....</i>	<i>161</i>
<i>Recipe 38: Entering Docker containers</i>	<i>161</i>
<i>Entering Config Server container.....</i>	<i>162</i>
<i>Entering Customer Enquiry container.....</i>	<i>162</i>
<i>Entering Enquiry Broadcast container.....</i>	<i>162</i>
<i>Pushing Docker images to image registry</i>	<i>163</i>
<i>Repository creation strategy.....</i>	<i>163</i>
<i>Recipe 39: Creating a repository in AWS ECR.....</i>	<i>163</i>
<i>Creating repository using AWS Management Console.....</i>	<i>164</i>

Creating repository using AWS CLI	164
Recipe 40: Authenticating to ECR	165
Recipe 41: Pushing images to ECR repository	166
Pushing Customer Enquiry image	167
Pushing Enquiry Broadcast image	167
Pushing Config Server image	168
Viewing images in the ECR repository	168
Conclusion	168
Points to remember	169
Questions	169
6. Pipeline Automation for CI/CD	171
Introduction	171
Structure	171
Objectives	172
Introduction to CI and CD pipelines	172
Benefits of CI/CD pipelines	173
GitLab CI/CD	174
Pipeline terminology	175
.gitlab-ci.yml file	175
Runners	175
Jobs	175
Stages	175
Building GitLab pipelines	176
Recipe 42: Creating a .gitlab-ci.yml file	176
Recipe 43: Viewing pipeline execution	179
Recipe 44: Enhancing build stage to generate artifact	180
Recipe 45: Adding a stage to generate Docker image of the artifact	182
Recipe 46: Enhancing test stage to test the Docker image	184
Recipe 47: Enhancing deploy stage to push the Docker image to Amazon ECR	186
Recipe 48: Adding rules in the deployment job to only execute the job conditionally ...	188
Linting	189

<i>Recipe 49: Adding a linting stage to check code style violations</i>	190
Scheduling GitLab pipelines	192
<i>Recipe 50: Scheduling a GitLab pipeline</i>	192
<i>Recipe 51: Using external variables to control pipeline execution</i>	194
Conclusion	197
Points to remember	197
Questions	198
7. Microservices Orchestration	199
Introduction	199
Structure	199
Objectives	200
Orchestration of microservices	200
Benefits of orchestration platforms	201
Load balancer	202
<i>AWS Application Load Balancer</i>	203
<i>Recipe 52: Creating AWS Application Load Balancer</i>	203
<i>Creating Application Load Balancer using AWS CLI</i>	203
Amazon Elastic Container Service	205
<i>Amazon ECS cluster</i>	205
<i>AWS Fargate</i>	205
<i>Recipe 53: Creating Amazon ECS cluster</i>	206
<i>Creating ECS cluster using AWS Management Console</i>	206
<i>Creating ECS cluster using AWS CLI</i>	207
<i>Amazon ECS task definition</i>	208
<i>Recipe 54: Creating Amazon ECS task definitions</i>	209
<i>Creating Config Server ECS task definition</i>	209
<i>Creating Customer Enquiry ECS task definition with JSON</i>	217
<i>Creating Enquiry Broadcast ECS task definition with JSON</i>	221
<i>Amazon ECS service</i>	221
<i>Recipe 55: Creating Amazon ECS Services</i>	221
<i>Creating Config Server ECS service</i>	222

<i>Creating Customer Enquiry ECS service</i>	226
<i>Creating Enquiry Broadcast ECS service</i>	228
<i>Recipe 56: Validating Application Load Balancer details</i>	229
<i>Recipe 57: Creating target group using AWS CLI</i>	231
<i>Recipe 58: Adding an HTTP listener to an ALB with default rule using AWS CLI</i>	232
<i>Recipe 59: Adding a conditional rule to a listener</i>	233
<i>Recipe 60: Validating Application Load Balancer details using AWS CLI</i>	234
Conclusion	235
Points to remember	236
Questions.....	236
8. Auto Scalability, High Availability, and Disaster Recovery	237
Introduction	237
Structure	237
Objectives	238
Auto scalability.....	238
<i>Automatic ECS scaling</i>	239
<i>Recipe 61: Auto scaling ECS services with target tracking policies</i>	239
<i>Recipe 62: Auto scaling ECS services with schedule scaling</i>	242
<i>Handling scale-in events gracefully</i>	247
<i>Recipe 63: Handling graceful shutdown in container</i>	247
High availability.....	249
<i>Highly available database</i>	250
<i>Recipe 64: Amazon RDS for database high availability</i>	250
<i>Highly available ECS service</i>	253
<i>Recipe 65: Amazon ECS for high availability</i>	254
<i>Recipe 66: Enhance load balancer to support multi-AZs</i>	255
Disaster recovery.....	256
<i>Recipe 67: Configure Amazon Route 53 for disaster recovery</i>	256
Conclusion	261
Points to remember	262
Questions.....	262

9. Cloud Security	263
Introduction	263
Structure	263
Objectives	264
Identity and access management.....	264
IAM policy	265
<i>Recipe 68: Creating IAM policy for accessing S3 bucket.....</i>	<i>267</i>
IAM group.....	270
<i>Recipe 69: Creating an IAM user group with permissions on S3 buckets</i>	<i>270</i>
IAM user	272
<i>Recipe 70: Creating an IAM user</i>	<i>272</i>
IAM role	274
<i>Recipe 71: Creating an IAM role for accessing S3 bucket</i>	<i>275</i>
Data protection	277
AWS Key Management Service	278
<i>Recipe 72: Creating customer managed key.....</i>	<i>279</i>
<i>Recipe 73: Using KMS key to encrypt S3.....</i>	<i>280</i>
AWS Secrets Manager	282
<i>Recipe 74: Storing a secret in Secret Manager.....</i>	<i>282</i>
<i>Recipe 75: Accessing secret from Secrets Manager in AWS ECS</i>	<i>284</i>
<i>Recipe 76: Enhance ECS execution role to access Secret Manager.....</i>	<i>285</i>
Network security.....	288
Virtual Private Cloud.....	288
<i>Recipe 77: Creating custom VPC using AWS CLI.....</i>	<i>288</i>
<i>Recipe 78: Creating subnets in a VPC using AWS CLI.....</i>	<i>289</i>
Security groups	291
<i>Recipe 79: Creating security group using AWS CLI.....</i>	<i>291</i>
<i>Recipe 80: Describing security group using AWS CLI</i>	<i>292</i>
<i>Recipe 81: Adding a rule to a security group.....</i>	<i>293</i>
AWS Web Application Firewall	295
<i>Recipe 82: Creating Web ACL with rules</i>	<i>296</i>
<i>Recipe 83: Associating Web ACL with ALB.....</i>	<i>299</i>
Conclusion	300

Points to remember	301
Questions.....	302
10. Observability.....	303
Introduction	303
Structure	303
Objectives	304
Observability	304
<i>Benefits of observability.....</i>	<i>304</i>
Logging.....	305
Container logging.....	306
AWS CloudWatch Logs.....	307
<i>Recipe 84: Creating a log group</i>	<i>307</i>
<i>Recipe 85: Instrumenting applications to push logs to CloudWatch.....</i>	<i>309</i>
<i>Recipe 86: Viewing logs in CloudWatch</i>	<i>310</i>
AWS CloudTrail	312
<i>Recipe 87: Creating trail with AWS CloudTrail</i>	<i>313</i>
<i>Recipe 88: Viewing CloudTrail Events.....</i>	<i>315</i>
Monitoring	316
CloudWatch Metrics	317
Custom CloudWatch metrics	317
<i>Recipe 89: Creating a custom metric.....</i>	<i>318</i>
CloudWatch alarms	322
<i>Recipe 90: Creating a CloudWatch Alarm.....</i>	<i>322</i>
<i>Recipe 91: Triggering and viewing an alarm</i>	<i>326</i>
CloudWatch dashboards	329
<i>Recipe 92: Creating custom CloudWatch dashboard</i>	<i>329</i>
Tagging	332
<i>Recipe 93: Adding tag to S3 bucket using AWS CLI</i>	<i>333</i>
<i>Recipe 94: Adding tag to ECS cluster using AWS CLI.....</i>	<i>334</i>
<i>Recipe 95: Adding tag to AWS RDS using AWS CLI</i>	<i>335</i>
<i>Recipe 96: Creating cost report for production resources.....</i>	<i>336</i>
Conclusion	338
Points to remember	338

Questions.....	339
11. Infrastructure Automation with IaC.....	341
Introduction	341
Structure	341
Objectives	342
Introduction to Infrastructure as Code	342
Terraform.....	344
<i>Terraform lifecycle</i>	344
<i>init</i>	345
<i>plan</i>	345
<i>apply</i>	345
<i>destroy</i>	345
<i>Terraform provider</i>	346
<i>Terraform configuration files</i>	347
<i>Terraform module</i>	348
<i>Terraform state</i>	348
Getting started with Terraform	348
<i>Terraform installation</i>	349
<i>Recipe 97: Installing Terraform</i>	349
<i>Installing Terraform on Windows</i>	349
<i>Installing Terraform on Linux (Ubuntu 20.04 LTS)</i>	349
<i>Setting the environment</i>	350
Writing Terraform scripts.....	351
<i>Recipe 98: Creating first Terraform script</i>	351
<i>Recipe 99: Enhancing Terraform script to create an S3 resource</i>	353
Provisioning infrastructure with Terraform.....	355
<i>Recipe 100: Executing Terraform script to provision an S3 resource</i>	355
Importing existing infrastructure in Terraform	359
<i>Recipe 101: Importing S3 bucket resource in Terraform</i>	360
Conclusion	363
Points to remember	363
Questions.....	363
Index	365-374

CHAPTER 1

Microservices and Cloud

Introduction

This is an era in which all greenfield application developments are done using microservice architecture, and going cloud is the strategic direction every organization is moving towards.

In this chapter, we will learn the importance of microservice architecture and how cloud elasticity is the go-to technology for leveraging the maximum benefits out of a microservice and the importance of Java frameworks in building enterprise grade microservices.

Structure

In this chapter, we will discuss following topics:

- Importance of microservices
- Importance of cloud
- Using the Java programming language
- Java frameworks for microservices
- Configuring cloud services

Objectives

By the end of this chapter, we will learn the benefits of developing an application using microservice architecture compared to a monolithic architecture, as well as the features provided by cloud infrastructure and how they overcome the challenges that are present in the on-premise deployments. We will also learn the importance of Java as a programming language for developing enterprise grade software applications, the Java frameworks available in the market for developing microservices and why Spring Boot is the most preferred option amongst them. In the end, we will learn how to configure different AWS services that will be required to begin the development of the microservices discussed in detail in later chapters of this book.

Importance of microservices

Microservices have evolved to overcome the issues and challenges faced by the monolithic applications, be it in any phase of the software development lifecycle, development, testing, build, deployment, or maintenance. Although microservices require some extra efforts, they provide many advantages in the larger extent.

The word **micro** means granular, that is, breaking the complete large requirement into small parts and then developing, building, and deploying them as services independent of each other, eventually orchestrating them in tandem to fulfill the complete business requirement. Here, granularity is a subjective term and how granular a service should be, is dependent on case-to-case basis and on the actual business functionality. In most of the cases, the functionality that is subject to change together in the future, should be part of the same microservice.

Figure 1.1 illustrates the difference between a monolithic application and microservices:

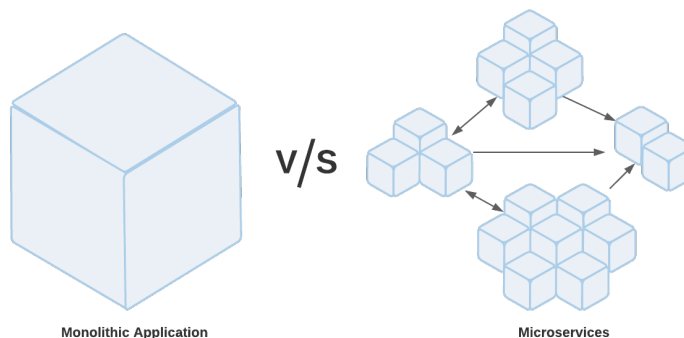


Figure 1.1: Monolithic versus Microservices

The benefits of developing an application using a microservice architecture compared with the overwork that needs to be done in different phases of **software delivery life cycle (SDLC)**, are explained as follows.

Development

The benefits and overwork in the development phase are as follows.

Benefits:

- Every microservice can be developed independently of each other.
- Different teams can work together to develop different microservices.
- If required, the microservices can be developed using different technology stacks.

Overwork:

- Multiple code repositories are to be created and maintained, one for each microservice.

Testing

The benefits and overwork in the testing phase are as follows:

Benefits:

- Every microservice can be independently tested by the testing team.
- Test cases can be easily created, since the scope of a microservice is limited compared to the complete functionality.

Overwork:

- Separate integration tests are to be created to test the overall business functionality.
- Multiple test environments might have to be created, if required.

Deployment

The benefits and overwork in the deployment phase are as follows:

Benefits:

- Build pipelines are simple and can be created independently for each microservice.
- Can leverage container and cloud infrastructure.
- **Independent Infrastructure as Code (IaC)** scripts can be created, which are simple and can reuse common templates.

Overwork:

- Microservices need to be carefully version controlled to resolve the interdependence on each other.

- Complex orchestration is required to build coordination between multiple microservices.

Management

The benefits and overwork in the management phase are as follows:

Benefits:

- Individual microservices can be upgraded or modified without impacting other services.
- Easier to debug and troubleshoot.
- Application health can be monitored at a microservice level.

Overwork:

- As more than one instance of each microservice can be live at a time, there is a need for complex observability patterns to get the complete insight of the whole application.

Additional advantages of application development using microservice architecture are:

- The design is highly scalable, and more microservices can be easily added later if required.
- Faster time-to-market as microservices are independently developed, tested, and deployed and the activities can be done in parallel.
- A microservice can be shared with different business modules, which makes it reusable.
- An individual microservice can be scaled in case of heavy load, without impacting others, which has a straight cost and performance benefit.

Thus, the benefits of a microservice architecture are huge compared to the overwork that is required to be done.

It is crucial to acknowledge that adopting a microservices architecture introduces new complexities. Communication between microservices becomes paramount, necessitating the implementation of robust and efficient mechanisms such as REST, gRPC APIs or message queues. Monitoring and managing the entire ecosystem of microservices demand more sophisticated tools and strategies such as service mesh. In conclusion, the microservices architecture is a powerful paradigm that addresses the limitations of monolithic applications. It promotes flexibility, scalability, and agility in software development. While it does present additional challenges, the benefits it provides, especially in handling large and complex systems, make it a preferred choice for modern application development.

Importance of cloud

Going cloud is the new norm; every agile company is either migrating their applications to cloud or building new applications in a cloud native way. Adopting cloud allows the enterprises to focus on their core competencies or businesses without bothering about any of the following:

- Hardware infrastructure
- Software licenses
- Security
- Patching
- Backups
- Disaster recovery, and so on.

Everything is taken care of by the cloud provider.

Apart from offloading the preceding tedious and tardy activities from the shoulders of an enterprise, cloud infrastructure provides them with the flexibility to choose from various options of compute and services, depending on their business needs and workloads. In return, enterprises also save costs, as they are only paying for services as per their usage, and there is no need to incur huge costs during off business hours or at the time of lean workloads.

There are multiple cloud providers in the market. Following are the ones that have majority of the market share:

- Amazon Web Services
- Microsoft Azure
- Google Cloud
- Oracle Cloud

Figure 1.2 features some of the cloud provider options:

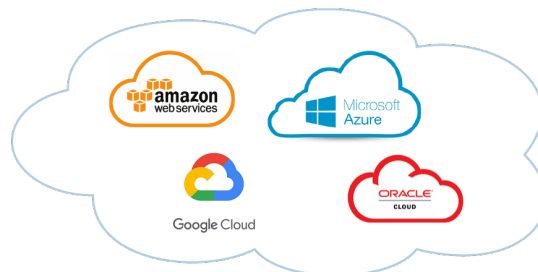


Figure 1.2: Cloud providers