

Concepts and Practices of DevSecOps

Crack the DevSecOps interviews

Ashwini Kumar Rath



www.bpbonline.com

First Edition 2024

Copyright © BPB Publications, India

ISBN: 978-93-55519-320

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 mother, the first architect of my words

About the Author

Ashwini Kumar Rath is an accomplished entrepreneur, prolific writer, and seasoned DevSecOps consultant with extensive computer programming experience. As the Director and CEO of Batoi, he has shaped its strategic vision and product development while extending his leadership to several tech companies he co-founded in India and internationally. Rath's educational background in theoretical physics and software management has fueled his successful transition into technology, culminating in the establishing of multiple tech startups. His academic interests encompass cloud computing, AI, and mathematical sciences, driving his contributions to numerous scholarly articles and business features in leading publications. As a frequent speaker at international forums, his expertise and insights are highly sought after. Committed to leveraging knowledge for socio-economic development, Rath also serves on various educational and business leadership boards.

About the Reviewers

- ❖ **Rakesh** is a seasoned technologist who has worked in the fast-paced technology field for 23 years. He gained expertise at Hewlett Packard, where he exhibited his passion for technology in positions like Technical Architect and Senior Software Engineer. Rakesh has been the Chief Technology Officer at In Time Tec for the past fifteen years, fulfilling his dream of helping clients worldwide overcome technology challenges. With a focus on IoT/Embedded Systems and Highly Scalable Cloud Solutions, his leadership experience ranges from creating low-level embedded software to creating high-performance cloud-based applications.

Rakesh strongly believes in the power of effective communication, analytical thinking, and team-building. Outside the boardroom, he remains current on the most recent technological advancements and supports open-source initiatives, showcasing his commitment to creativity and lifelong learning. Rakesh is always eager to collaborate on worthwhile initiatives with like-minded enthusiasts, highlighting the positive impact of technology on our daily lives. His diverse technological background reflects a commitment to expanding the possibilities of technology.

- ❖ **Ankit** is a senior cloud data architect with Google and has over a decade of progressive work experience in Cloud, Data, AI, and ethical machine learning practices. He works with customers to develop scalable, reliable, and performant data and machine learning platforms and has worked across companies like Amazon, CVS, Deloitte, and Infosys. He is a technical reviewer for various books focused on data governance, ethical AI, AI governance, Data and AI platforms, Python for AI, and other material related to the promotion of AI ethically and responsibly in society. He earned his master's from the esteemed Kelley School of Business and serves as the advisory board member for the Institute of Business Analytics, Indiana University Bloomington.

Acknowledgement

My most profound appreciation goes out to my beloved family and steadfast friends, whose constant support and motivation served as my guiding star throughout the creation of this book. In particular, my wife, Mita, deserves a special mention for her unwavering encouragement and patience during my perpetual travels and work, gently nudging me to complete each chapter.

I owe a significant debt of gratitude to BPB Publications. Their expertise and professional guidance were pivotal in transforming the raw manuscript into this finished work. It has been an enriching journey, enriched further by the diligent reviewers, technical experts, and editors whose input significantly enhanced the content and structure of this book.

My time in the tech industry, filled with the teachings and insights from my esteemed colleagues and peers from industry and academics, has been instrumental in my growth and understanding. Their generous contributions have immeasurably shaped the narratives and perspectives within these pages.

Last but not least, my sincere thanks to you, the readers, whose interest and support fuel the purpose of this work. Your enthusiasm is the heart of this endeavor, and your encouragement has been priceless. Thank you for making this book a reality.

Preface

Welcome to **Concepts and Practices of DevSecOps**!

As we find ourselves in the midst of a digital revolution, one thing has become evident – the importance of integrating security into our development and operational practices. This has given rise to the practice of DevSecOps, a discipline that builds upon the principles of DevOps with a sharpened focus on security.

This book aims to demystify the DevSecOps field, from its foundational concepts to the advanced practices and technologies that define it today. This comprehensive guide has been designed to equip you with the knowledge you need to excel in DevSecOps roles and to confidently face interviews that test your understanding and application of these principles.

The chapters in this book walk you through different aspects of DevSecOps, including Application Security, Infrastructure as Code, Containers and Security, Automation and Integration, and the Frameworks and Best Practices underpinning the discipline. As we journey into the world of DevSecOps together, we will explore how it plays a pivotal role in the current era of Digital Transformation.

Whether you are a DevOps engineer, project manager, product manager, software developer, or any professional seeking to fortify your understanding of DevSecOps, this book is a crucial resource. With a working knowledge of DevOps, you will find the content accessible and enlightening, helping you to contribute to or lead a DevSecOps team effectively.

Each chapter is structured to answer the pressing questions surrounding each topic, supplemented with practical use cases that bring the theoretical concepts to life. I have also included resources for further reading at the end of each section, allowing you to delve deeper into the subjects that interest you the most.

The world of DevSecOps is as fascinating as it is intricate. This book prepares you for your career progress in this field and ignites a passion for security's vital role in our ever-evolving digital landscape.

May this book serve as your compass, guiding you through the complex terrain of DevSecOps and helping you become an influential team member as you work together to build secure, efficient, and innovative solutions.

Chapter 1: Security in DevOps – It delves into the interweaving of security within DevOps culture, providing insights into the transformation of security measures within software systems management. It establishes a comprehensive framework for understanding DevSecOps, laying a foundation for the forthcoming chapters.

Chapter 2: Application Security – It offers an in-depth exploration of application security, discussing the intricate details of various application architectures and their inherent security aspects. Readers will gain a firm grasp of the tools and technologies that fortify applications, accompanied by a practical case study on constructing and sustaining an enterprise application.

Chapter 3: Infrastructure as Code – It acquaints the reader with different cloud platforms and essential infrastructure management tools instrumental in executing successful DevSecOps projects. Covering several leading vendors, it offers insights into tool selection for specific scenarios while focusing on scalability and change management for contemporary IT systems.

Chapter 4: Containers and Security – It offers a thorough understanding of security measures surrounding popular container technologies, emphasizing an understanding of various vulnerabilities. It examines solutions and processes for vulnerability management and discusses the optimal tools and techniques available.

Chapter 5: Automation and Integration – It introduces platforms and tools for comprehensive security management, particularly crucial in managing enterprise systems and large-scale software with substantial attack surfaces. It discusses a range of cloud security solutions, including CWPP, CSPM, CASB, and CNAPP, supplemented with a practical use case to demonstrate the substantial benefits of integration.

Chapter 6: Frameworks and Best Practices – It provides a detailed view of leading security frameworks and their management in the DevSecOps process, including audit, compliance, reporting, visualization, and threat modeling.

Chapter 7: Digital Transformation and DevSecOps – It discusses a lean approach to digital transformation projects with a spotlight on DevSecOps management. It touches upon cultural aspects, skills, roles, and responsibilities, managing technical liabilities, and establishing secure development practices.

Code Bundle and Coloured Images

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

<https://rebrand.ly/4c8758>

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

<https://github.com/bpbpublications/Concepts-and-Practices-of-DevSecOps>.

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. Security in DevOps.....	1
Introduction.....	1
Structure.....	1
Objectives	2
<i>Relooking at security operations</i>	<i>2</i>
<i>A DevOps cycle.....</i>	<i>2</i>
<i>Conventional SecOps with DevOps</i>	<i>3</i>
<i>Issues with conventional SecOps</i>	<i>4</i>
Shifting security left	4
Adopting DevSecOps: Key changes	5
<i>Lean process.....</i>	<i>6</i>
Agile versus DevSecOps: No contradiction	7
<i>Automation</i>	<i>7</i>
<i>Measurement.....</i>	<i>7</i>
<i>Ecosystem interoperability</i>	<i>8</i>
<i>Documentation of new and old ways</i>	<i>9</i>
Security controls	10
<i>Goals of security</i>	<i>10</i>
Documentation and security	11
Threat modelling and security policies.....	12
Infrastructure provisioning and security.....	13
<i>A high availability configuration</i>	<i>13</i>
<i>Managing infrastructure in the Cloud</i>	<i>14</i>
<i>Security intervention for infrastructure provisioning.....</i>	<i>14</i>
Code commit, release and security	15
<i>A programming framework.....</i>	<i>15</i>
<i>Automated security tools in software environments</i>	<i>17</i>
A use case: IoT application.....	18

Conclusion.....	20
Questions.....	21
2. Application Security.....	23
Introduction.....	23
Structure.....	23
Objectives	24
An app on the Cloud	24
<i>Delivery of Cloud services</i>	24
<i>Identity and access management</i>	25
<i>Metering and billing</i>	27
<i>Provisioning and resource management</i>	28
<i>Constituents of an app</i>	28
<i>App and workflow</i>	29
<i>Session management: Cookie and JSON web token</i>	30
<i>Encryption</i>	33
<i>Hash function</i>	34
<i>Public key infrastructure and secured socket layer</i>	35
<i>Microservices</i>	37
<i>High availability deployment and multi-instance deployment</i>	38
<i>Serviceful and serverless</i>	40
<i>Putting all together: A security perspective</i>	42
CI /CD pipeline and security	42
<i>Web application firewall</i>	44
<i>Vulnerability DBs, automation and monitoring</i>	46
<i>InfoSec as a service</i>	46
Low-Code, No-Code and RAD	48
<i>Business operation, workflow, and communication</i>	49
<i>Different techs and new ways of application development</i>	50
Application security.....	50
OWASP Top 10	51
<i>A01:2021 - Broken access control</i>	51

<i>A02:2021 - Cryptographic failures</i>	52
<i>A03:2021 - Injection</i>	52
<i>A04:2021 - Insecure design</i>	52
<i>A05:2021 - Security misconfiguration</i>	52
<i>A06:2021 - Vulnerable and outdated components</i>	53
<i>A07:2021 - Identification and authentication failures</i>	53
<i>A08:2021 - Software and data integrity failures</i>	53
<i>A09:2021 - Security logging and monitoring failures</i>	53
<i>A10:2021 - Server-side request forgery</i>	53
<i>SAN Top 25</i>	54
<i>Use case: Making a secure application</i>	54
Conclusion.....	55
Questions.....	55
3. Infrastructure as Code	57
Introduction.....	57
Structure.....	58
Objectives	58
Cloud infrastructure.....	58
Benefits of IaC in DevSecOps	62
IaC for DevSecOps in AWS.....	64
<i>Define our IaC with AWS CloudFormation</i>	64
<i>Set up CI/CD pipeline</i>	65
<i>Incorporate security controls</i>	66
<i>Use AWS Config for continuous compliance</i>	67
<i>Automated testing</i>	67
<i>Incorporate monitoring and logging</i>	67
<i>Use AWS Secrets Manager for managing secrets</i>	67
IaC for DevSecOps in GCP.....	67
<i>Define our IaC with deployment manager</i>	68
<i>CI/CD pipeline</i>	68
<i>Incorporate security controls</i>	70

Use Google Cloud Asset Inventory for continuous compliance.....	70
Automated testing.....	70
Incorporate monitoring and logging.....	70
Use Google Secret Manager for managing secrets.....	71
IaC for DevSecOps in Azure.....	71
CI/CD pipeline.....	73
Incorporate security controls.....	74
Use Azure Policy for continuous compliance.....	75
Automated testing.....	76
Incorporate monitoring and logging.....	76
Use Azure Key Vault for managing secrets.....	77
IaC for DevSecOps in a hybrid environment.....	77
Define our Infrastructure as Code.....	78
CI/CD pipeline.....	78
Incorporate security controls.....	80
Continuous compliance.....	81
Automated testing.....	81
Incorporate monitoring and logging.....	81
Secret management.....	81
IaC and DevSecOps with legacy system.....	81
Implementing IaC and DevSecOps with legacy systems.....	82
DevSecOps dashboard.....	83
Use case: Setup software environment.....	84
Conclusion.....	85
Questions.....	86
4. Containers and Security.....	87
Introduction.....	87
Structure.....	88
Objectives.....	88
Introduction to containers.....	89
Natural fit for microservices.....	90

Container technologies	90
<i>Overview of Docker</i>	90
<i>Introduction to Kubernetes</i>	91
<i>Other container orchestration tools</i>	92
Role of containers in DevSecOps	92
<i>Consistency and reproducibility</i>	92
<i>Isolation</i>	93
<i>Scalability and efficiency</i>	93
<i>Immutable Infrastructure</i>	93
Container security basics	94
<i>Container images</i>	94
<i>Storing and distributing securely</i>	96
<i>Container runtime</i>	96
<i>Container isolation</i>	96
<i>Least privilege</i>	96
<i>Security modules</i>	97
<i>Runtime vulnerability scanning</i>	97
<i>Host system</i>	97
<i>Orchestration and deployment</i>	97
<i>Role-based access control</i>	98
<i>Securing the control plane</i>	98
<i>Network policies</i>	98
<i>Importance of container security in DevSecOps</i>	99
<i>Challenges in container security</i>	100
Security in container lifecycle	101
<i>Secure container development</i>	101
<i>Secure container deployment</i>	102
<i>Secure container operations</i>	102
Container image security	103
<i>Importance of secure container images</i>	103
<i>Vulnerabilities in container images</i>	104
<i>Clair</i>	104

<i>Anchore Engine</i>	105
<i>Docker Security Scanning</i>	105
<i>Signing and verifying container images</i>	106
<i>Docker Content Trust</i>	106
<i>Notary</i>	106
<i>Portieris</i>	107
Runtime container security.....	107
<i>Container isolation mechanisms</i>	107
Namespaces	107
Control groups	108
Capabilities.....	108
<i>Monitoring and auditing container activity</i>	109
<i>Monitoring container activity with Fluentd</i>	109
<i>Log analysis with Elasticsearch and Kibana</i>	109
<i>Auditing container activity with Auditd and Falco</i>	110
<i>Detecting and responding to runtime threats</i>	110
Network security for containers.....	111
<i>Container network models</i>	112
Bridge networks	112
Host networks	112
Overlay networks.....	112
<i>Implementing network policies</i>	113
<i>Secure service discovery and communication</i>	114
Service discovery	114
Secure communication	114
Secrets management in containers.....	115
<i>Challenges of managing secrets in containers</i>	115
Ephemeral nature of containers	115
Scale	116
Immutable infrastructure.....	116
Secure strategies for storing and accessing secrets	118
Environment variables	118

<i>Secrets volume</i>	118
<i>Secrets management service</i>	119
<i>Tools for secrets management in containers</i>	119
<i>Docker secrets</i>	119
<i>Kubernetes Secrets</i>	119
<i>Vault by HashiCorp</i>	120
<i>Cloud secrets management services</i>	120
Best practices for container security in DevSecOps	121
<i>Following the principle of least privilege</i>	121
<i>Running containers as non-root user</i>	121
<i>Limiting container capabilities</i>	121
<i>Implementing fine-grained access control</i>	122
<i>Regularly updating and patching containers</i>	123
<i>Updating container images</i>	123
<i>Deploying updated containers</i>	123
<i>Monitoring for vulnerabilities</i>	124
<i>Using immutable containers</i>	124
<i>Automated security scanning and remediation</i>	124
<i>Automated security scanning during build</i>	124
<i>Continuous scanning</i>	125
<i>Automated remediation</i>	125
<i>Alerts and manual intervention</i>	125
<i>Integrating security into the CI/CD pipeline</i>	125
<i>Image scanning during build</i>	125
<i>Static code analysis</i>	126
<i>Security policy enforcement</i>	127
Case studies of container security in DevSecOps	127
<i>Case study: Adobe</i>	127
<i>Case study: Shopify</i>	128
Conclusion	128
Questions	129

5. Automation and Integration	131
Introduction.....	131
Structure.....	132
Objectives	132
Automating integration workflows.....	133
Policy as Code.....	134
Monitoring as code.....	138
Security as code	140
<i>Automated security checks.....</i>	<i>140</i>
<i>Infrastructure security</i>	<i>142</i>
<i>Secure defaults.....</i>	<i>143</i>
<i>Authentication and authorization.....</i>	<i>145</i>
<i>Identity and access management tools.....</i>	<i>145</i>
<i>Multi-factor authentication</i>	<i>146</i>
<i>Single sign-on</i>	<i>146</i>
<i>Identity federation.....</i>	<i>146</i>
<i>Secrets management.....</i>	<i>146</i>
Cloud security solutions.....	147
<i>Cloud workload protection platforms</i>	<i>147</i>
<i>Cloud security posture management.....</i>	<i>147</i>
<i>Cloud access security brokers</i>	<i>148</i>
Supply chain and risks	150
<i>Potential vulnerabilities</i>	<i>151</i>
<i>Possible exploits.....</i>	<i>151</i>
<i>Mitigation strategies</i>	<i>152</i>
Automating integration workflows challenges and best practices.....	153
Use case: Integrations	154
Conclusion.....	158
Questions.....	158
6. Frameworks and Best Practices	161
Introduction.....	161

Structure.....	161
Objectives	162
Risks and compliance	162
Security frameworks.....	164
<i>ISO/IEC 27001</i>	165
<i>National Institute of Standards and Technology Cybersecurity Framework</i>	166
<i>Center for Internet Security Controls</i>	167
<i>Payment card industry data security standard</i>	169
<i>Control objectives for information and related technologies</i>	170
<i>Health Insurance Portability and Accountability Act</i>	171
<i>System and Organization Controls 2</i>	172
<i>Working with different frameworks</i>	173
Compliance as code and its importance	177
Understanding security audit workflows.....	179
Threat modeling	182
<i>STRIDE</i>	182
<i>Process for attack simulation and threat analysis</i>	182
<i>DREAD</i>	182
<i>OCTAVE</i>	182
<i>Attack trees</i>	183
CSA's six pillars of DevSecOps	183
Compliance and risk management for our IoT application.....	185
Conclusion.....	187
Questions.....	188
7. Digital Transformation and DevSecOps	189
Introduction.....	189
Structure.....	189
Objectives	190
The nature of digital transformation	190
DevSecOps: Roles, responsibilities, and skillsets	192
Cultivating a new culture: The human element	195

Collective responsibility	196
Open communication and collaboration	196
Pragmatic implementation and continuous learning	196
Automation and empowerment.....	196
Threat investigation and embracing failure	197
Open-source software balancing opportunities and challenges	198
Opportunities presented by open-source software	198
Innovation and flexibility.....	198
Reduced costs	198
Availability of high-quality tools	198
Driving innovation in DevSecOps	199
Challenges of open-source software	199
Security risks.....	199
Dependency management	199
Quality and maintenance variability.....	200
Technological liability.....	200
Towards successful open-source initiatives in DevSecOps.....	200
The journey towards cloud-native capabilities.....	201
Conclusion.....	203
Questions	204
Index	207-218

CHAPTER 1

Security in DevOps

Introduction

While DevOps promises a great deal to different stakeholders, security has long been an exclusive focus for any software development team. Traditionally, companies outsource the security audit to an external agency or keep the stuff separate for a group of security experts. Moreover, all security management functions start after a software version is deployed into production. Such a scenario experiences a few to and fro exchanges that include tool-based scanning and manual testing by the security team on the one hand and the resulting effort of troubleshooting by the software or IT team on the other. Consequently, it significantly increases the time of a DevOps cycle and nullifies the fundamental purpose of DevOps, which commits to faster software delivery.

Let us dive into the first chapter.

Structure

In this chapter, we will discuss the following topics:

- Relooking at security operations
- Shifting security left
- Adopting DevSecOps: Key changes

- Security controls
- Documentation and security
- Threat modelling and security policies
- Infrastructure provisioning and security
- Code commit, release and security
- A use case: IoT application

Objectives

This chapter offers an insight into the security aspects of software systems and their management from the perspective of DevOps culture. We discuss the significant technological advancements, tools, and practices that have changed how we treat and incorporate security into the scheme of things. While the chapter serves as a solid introduction to DevSecOps, it also builds a broader framework for the later chapters.

Relooking at security operations

With the rapid adoption of cloud computing and remote working, the IT infrastructure is moving out of the cozy confines of office networks. Security has been more important than ever before. In this section, we shall relook at the **security operations (SecOps)** and how it fits into DevOps. Before we proceed further, let us review what a DevOps cycle looks like.

A DevOps cycle

It is famously illustrated with an infinity loop (*Figure 1.1*) broadly running over six stages:

- **Design:** You design new software, an improvement to existing software, or even a modification to the software to align with requirements.
- **Code:** The phase includes programming activities of coding, compiling (when needed) and testing the software units.
- **Integrate:** While different software developers work on different units, integration does the process of merging the changes into a codebase that will function as the designated software system.
- **Deploy:** After integration, the software system needs to be deployed at one or multiple server locations as per the deployment architecture.
- **Operate:** Users start using the system, they can be a selected user group (for doing a pilot) or the actual end users for whom the software is designed.
- **Monitor:** While the software is in use, its accuracy, usability, and performance must be monitored, and adequate feedback is gathered for the next cycle in the loop.

The 'monitor' stage of a cycle feeds the 'design' stage of the next loop cycle, thus creating an infinitely evolving software system's lifecycle. Every business owner or manager loves it, but they would expect the software to have minimal defects yet great security controls. They would also expect that each cycle should be as fast as possible to match the business goals and should lower the effective cost of software development and IT operation management.

It should be noted that different authors or practitioners describe the loop in slightly different ways. However, they fall broadly into the ones that we have described above.

The function of the DevOps infinite Loop is illustrated in *Figure 1.1*:

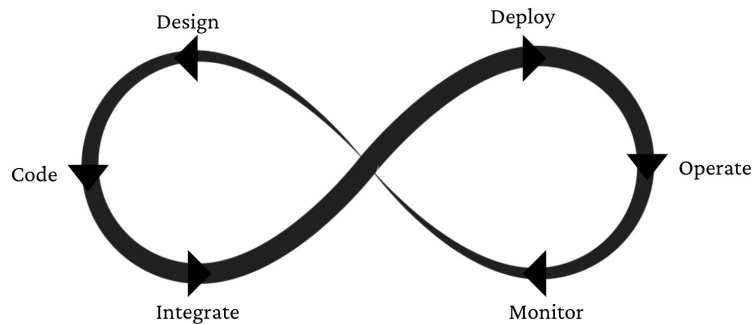


Figure 1.1: DevOps infinite loop

Conventional SecOps with DevOps

Let us cite an example to understand the traditional method of SecOps or security operations. Once, a software agency, which had recently adopted DevOps, was tasked with developing an application for its customer. The customer had their own chief security officer and chief risk officer. During one of the project review meetings, these two officers wanted to understand the software development cycle to eliminate process-related risks and have a transparent process for software development and management. The agency had automated the infrastructure provisioning on the Cloud and CI/CD pipeline. They could complete one DevOps cycle in less than a day. However, after each software release (software moving to production), they will refer to the security vendor, who will do a round of penetration testing and provide the list of findings (security issues observed) for the software development team to review and respond. The software team would do the following:

- If a finding is a false positive, they will ignore it.
- If a finding is of a high-risk category, they will fix it.
- If a finding refers to a low-risk category, they may fix or ignore it.

After the above tasks, the software team will respond to the security team. The security team will do another round of penetration testing. Here, the security team may add them