# Demystifying DevSecOps in AWS

*Achieve operational excellence in the cloud with DevSecOps*

**Picklu Paul**

www.bpbonline.com

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

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

First published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

**UK | UAE | INDIA | SINGAPORE**

ISBN 978-93-55515-995

www.bpbonline.com

**Dedicated to**

*My beloved parents:*
***Shri Pradip Chandra Paul***
***Smt. Silpi Paul***
*&*
*My beautiful wife,* ***Sumita***

# About the Author

**Picklu Paul** is a dynamic and forward-thinking cybersecurity leader based in Singapore. With a decade of diverse experience across multiple IT domains, he brings a fresh perspective and a passion for pushing the boundaries of Cybersecurity.

With a strong foundation in both DevSecOps and AWS security, Picklu has quickly risen to prominence as a thought leader in the industry. His innovative contributions and dedication to staying ahead of evolving threats have earned him respect and recognition among peers.

When not immersed in the world of cybersecurity, Picklu is a passionate chess player and an occasional billiards enthusiast. His diverse interests reflect a well-rounded personality, always eager for new challenges and experiences.

His latest publication, '**Demystifying DevSecOps in AWS**: *Achieve operational excellence in the cloud with DevSecOps*', reflects his commitment to empowering the next generation of cybersecurity leaders. It encapsulates Picklu’s vision for achieving DevSecOps excellence in AWS environments and underscores his status as a prominent leader in the DevSecOps realm.

# About the Reviewers

- **Srinivas A Vaddadi** is a forward thinking expert in Cloud and DevSecOps. With a profound understanding of cutting-edge technologies and a passion for innovation, Srinivas has emerged as a prominent figure in the industry.

  Srinivas has showcased a remarkable ability to navigate the complex landscape of cloud computing and security. His expertise in designing and implementing robust cloud architectures, coupled with a deep understanding of DevSecOps principles, has enabled organizations to achieve enhanced efficiency, scalability, and security in their operations. With a solid educational foundation in computer science, Srinivas embarked on a journey of continuous learning and professional growth. Their relentless pursuit of knowledge and commitment to staying at the forefront of industry advancements has earned them recognition as a thought leader in the Cloud and DevSecOps space. Srinivas's contributions extend beyond their technical expertise. They are a seasoned leader and mentor, guiding and inspiring cross-functional teams to embrace agile methodologies and cultivate a culture of continuous improvement. Furthermore, Srinivas's passion for sharing knowledge and contributing to the community is evident through their active participation in conferences, workshops, and industry forums. As a sought-after speaker and published author, he has disseminated valuable insights and best practices, empowering professionals worldwide to embrace the transformative power of Cloud and DevSecOps.

  In summary, Srinivas A Vaddadi is a visionary professional in the field of Cloud and DevSecOps, leading the charge towards innovative and secure cloud solutions. Their expertise, leadership, and commitment to advancing the industry make them an invaluable asset to any organization seeking to harness the full potential of the cloud while prioritizing security and operational excellence.

- **Quentin Scheffer**, a Cyber Security professional who spent the last decade designing, building and securing different on-prem and cloud environments.

  He has been working for family offices, AI/ML startups, consultancies and banks, building and securing everything he could get his hands on. From networks to servers to clouds to applications, taking on challenges each step of the way and learning from them.

  Outside of work he does martial arts, climbing and parkour to keep his body and mind healthy. These days, he is working for an investment management company as a cloud security expert. Currently learning a new language, he is deeply interested in Japanese culture, traditions and history.

# Acknowledgement

Bringing this book to life has been a profound journey, and I owe my deepest gratitude to the many individuals and forces that have shaped it.

First and foremost, I want to express my heartfelt thanks to my family – my parents, my wife, and my brother. All your unwavering support, sacrifices, and encouragement have been my constant source of strength, without which I could have never completed this book.

I would like to acknowledge the numerous mentors and colleagues who have shared their knowledge and experiences with me throughout my career. Your insights have enriched my understanding of cybersecurity and DevSecOps, and I am deeply grateful for your contributions.

I extend my appreciation to the dedicated team at BPB Publication, whose expertise and commitment have transformed this project from a vision into reality. Your guidance and meticulous efforts have been instrumental in shaping the content within these pages.

To the readers of this book, I extend my warmest appreciation for choosing to explore the world of DevSecOps in AWS with me. Your curiosity and pursuit of knowledge drive the continual evolution of our field.

Lastly, I pay tribute to the broader cybersecurity community, a collective of individuals and organizations dedicated to securing our digital future. Your dedication to enhancing cybersecurity practices inspires us all.

This book stands as a collaborative effort, and it is a reflection of the support and encouragement I have received from so many. I am truly grateful for the opportunity to contribute to the discourse on DevSecOps, and I look forward to the continued journey of learning and growth in the dynamic world of cybersecurity.

# Preface

I am humbled to introduce this book, **Demystifying DevSecOps in AWS**: Achieve operational excellence in the cloud with DevSecOps. This book is the result of extensive dedication and research, driven by the singular goal of equipping professionals to navigate the intricate realm of cybersecurity AWS environments.

My journey through the dynamic world of cybersecurity, DevSecOps, and AWS has been a continuous learning experience. Throughout this book, I aim to share the knowledge and insights I've gathered, not as an expert but as someone passionate about these subjects and eager to learn alongside you.

This book is not just a guide; it's a collaborative effort to demystify DevSecOps principles, AWS security practices, and their real-world implementation. Whether you're a seasoned AWS practitioner, a cybersecurity expert, a DevOps enthusiast, or an academic researcher, this book empowers you to master DevSecOps in AWS. It seamlessly integrates security into your development and operations workflows, fostering collaboration and automation to safeguard your applications and data while achieving operational excellence in the cloud.

**Chapter 1: Getting Started with DevSecOps –** This chapter explains the fundamentals and importance of DevSecOps in the IT industry. We explore the evolution from traditional Waterfall to Agile methods, the transition from Agile to DevOps, and the critical need for DevSecOps.

**Chapter 2: Infusing Security into DevOps –** This chapter dives into the practicalities of infusing security into various stages of the DevOps pipeline. We explore the stages of a typical DevOps pipeline and discuss the additional steps required to embed security seamlessly and strategies for building security into the pipeline.

**Chapter 3: DevSecOps Process and Tools –** This chapter dives into DevSecOps processes and the relevant tools necessary for their implementation. We cover an array of security measures at each phase of the software development lifecycle, including software planning, design, build, testing, deployment, operation, and more. It's a comprehensive guide to integrating security into every aspect of your DevSecOps pipeline.

**Chapter 4: Build Security in AWS Continuous Integration –** Our journey in AWS begins with Chapter 4, where we explore securing the **Source Code Management** (**SCM**) phase. We introduce AWS services like CodeBuild and third-party solutions for **Static Code Analysis** (**SCA**) and **Static Application Security Testing** (**SAST**). This chapter focuses on ensuring code security right from the start.

**Chapter 5: Build Security in AWS Continuous Deployment –** This chapter continues our exploration of AWS by discussing securing the Continuous Deployment phase. We dive into AWS services such as CodePipeline and explore security measures during testing, including **Dynamic Application Security Testing** (**DAST**) and penetration testing. It's all about securing the deployment process within AWS.

**Chapter 6: Secure Auditing, Logging, and Monitoring in AWS –** In this chapter, we shift our focus to auditing, logging, and monitoring applications within AWS. AWS services like CloudTrail, CloudWatch, and GuardDuty become our allies in safeguarding against threats. We also delve into third-party SIEM solutions from the AWS Marketplace.

**Chapter 7: Achieving SecOps in AWS –** This Chapter emphasizes continuous security checks, a crucial stage in DevSecOps. We look at vulnerability management using AWS Inspector, AWS Security Hub, bug bounty programs, perimeter security with AWS WAF, and incident response using Amazon Detective.

**Chapter 8: Building a Complete DevSecOps Pipeline in AWS –** This chapter provides a comprehensive understanding of implementing the DevSecOps pipeline within AWS. We examine a sample pipeline implementation that utilizes various AWS services, bridging the gap between theory and practical implementation.

**Chapter 9: Exploring a Real-world DevSecOps Scenario –** This chapter offers a practical use case, immersing you in a real-world DevSecOps problem. By studying this scenario, you gain hands-on experience, witness the DevSecOps pipeline in action, and learn how to design and implement one for your organization.

**Chapter 10: Practical Transformation from DevOps to DevSecOps Pipeline –** This Chapter guides organizations in transforming their DevOps pipelines into DevSecOps pipelines through practical use case by allowing you to assume the role of a DevSecOps engineer.

**Chapter 11: Incorporating SecOps to Complete DevSecOps Flow –** The final chapter focuses on incorporating SecOps into the pipeline to achieve a complete DevSecOps flow. We explore security tools and practices integration, discuss common challenges, and provide strategies for overcoming them.

As you embark on this journey, remember that in the ever-evolving world of cybersecurity, we are all perpetual students. I encourage you to approach this book with curiosity, ask questions, experiment, and, above all, embrace the spirit of continuous learning. Together, we'll explore the multifaceted world of DevSecOps in AWS and work towards achieving excellence while remaining humble in the face of its complexities.

Thank you for joining me on this transformative expedition.

# Code Bundle and Coloured Images

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

## https://rebrand.ly/4c685vz

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Demystifying-DevSecOps-in-AWS**. 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.

### 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

# CHAPTER 1
# Getting Started with DevSecOps

## Introduction

Over the past three decades, software development has undergone a significant evolution to meet the evolving needs of businesses. In the earlier days, software primarily played a supporting role, providing tools and utilities to assist various business functions. However, in today's landscape, software has transformed into a crucial enabler or even the core of businesses themselves. This transformation is driven by the recognition that software solutions can provide a competitive advantage and drive value for organizations.

The importance of this shift cannot be overstated. Software has become the backbone of modern businesses, empowering them to streamline operations, enhance customer experiences, and drive innovation. It enables businesses to differentiate themselves in the market, optimize internal processes, and adapt to the ever-changing demands of customers and the industry. As businesses rely more heavily on software, it has become essential for software development techniques to evolve in tandem.

To support this transformation, software development techniques have evolved to match the complexity and agility required in the modern business landscape. These techniques enable businesses to respond swiftly to changing market dynamics and customer needs. The evolution of software development techniques ensures that businesses can deliver innovative solutions efficiently and effectively. It allows for faster time-to-market, iterative development, and continuous improvement. This agility in software development enables businesses to stay competitive and thrive in today's fast-paced and demanding business environment.

# Structure

The following are the topics to be discussed in this chapter:

- Evolution from Waterfall to Agile methods
- Transforming from Agile to DevOps
- Need for DevSecOps
- Common Aspects of DevOps and DevSecOps
- Difference between DevOps and DevSecOps

# Objectives

This chapter aims to explore the evolution of software development methodologies from the traditional waterfall model to agile methodologies and subsequently to DevOps and DevSecOps practices. The objective is to understand the importance of this progression in meeting the dynamic needs of modern businesses. The chapter will begin by examining the limitations of the waterfall model, such as lengthy development cycles and limited adaptability, which led to the emergence of agile methodologies. It will delve into the principles and benefits of agile methodologies like Scrum and Kanban, highlighting their focus on flexibility, collaboration, iterative development, and faster time-to-market. Next, the chapter will delve into the transformation from agile methodologies to DevOps practices. It will explore how DevOps integrates development and operations teams, emphasizing collaboration, automation, and continuous delivery. The benefits of DevOps, including improved efficiency, faster software delivery, and enhanced collaboration between teams, will be discussed.

Lastly, the chapter will address the growing importance of security in software development and the emergence of DevSecOps. It will outline how DevSecOps integrates security practices into the software development lifecycle from the beginning, ensuring proactive security measures and protecting sensitive data and systems.

By the end of this chapter, readers will have a comprehensive understanding of the evolution from the waterfall model to agile methodologies, and further to DevOps and DevSecOps practices. They will grasp the significance of this progression in meeting the complexity, agility, and security requirements of modern businesses.

# Evolution from Waterfall to Agile methods

Most of the software development processes in previous decades were centered around linear and sequential methodologies like Waterfall. All the project activities were supposed to be completed sequentially in a particular order. These activities were grouped into different phases. All the activities in one phase should be completed before moving to the next phase. An example of this is the Waterfall methodology, where the software development life cycle is divided into sequential phases. In the first phase, the team would gather and document all the requirements for the application from the stakeholders. This includes their technical specifications, budget, timeline, risks, dependencies, success metrics, and any other details required for the completion of the project. Then comes the design phase, where the team defines how to technically implement the business logic including layout designs, data models, and scenarios. In the next phase, developers write the source code to implement the models, business logic, and service integrations based on the requirements and specifications. The application is rigorously tested, for example, Unit testing, Integration Testing, Regression Testing Performance Testing, etc before going into production for end customers to ensure no errors are introduced into the final product and all requirements have been met, resulting in a good user experience. Product managers supply the testing team with design documents and usecase scenarios for creating the test cases. Minor bugs which are detected during the testing phase are fixed. However, in case of major faults, the teams have to restart with the first phase. Finally, the last phase is the deployment of the application, which will however need to be maintained and updated over time. All the change requests or feature requests coming from the users will be handled by the teams in the subsequent releases of new versions of the software. Let us look into the different phases of Waterfall method as shown in the following figure:

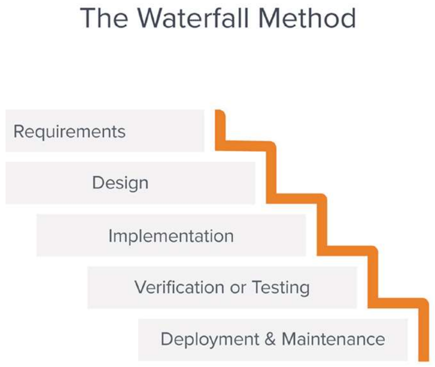


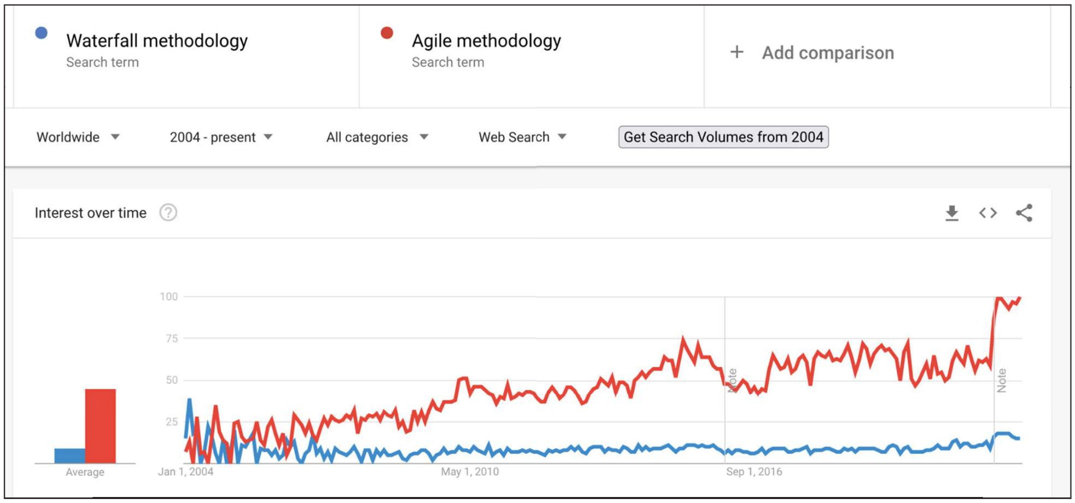
***Figure 1.1:*** *Phases of Waterfall Methodology*

Another profound effect of using the waterfall model was the creation of **silos**. Silos in software development refer to isolated groups or departments that operate independently, with limited communication and collaboration between them, leading to inefficiencies and challenges in the development process. When large firms embrace this model and expand, they tend to develop departments for each of the phases. Because they have different objectives, deliverables, and goals, each of those departments tends to work in isolation from the others. At the end of one phase, the product is "thrown over the wall" to the team in charge of the next step. After that, the same department frequently goes on to the next project, leaving little time to support the team working on the next phase. The most common side effect of phases and "silos" is an increase in administration and documentation effort. It is fairly unusual for medium to large projects to spend one-third or even half of their budget on paperwork and management. Because stages and silos restrict team communication, it must be compensated for by increasing documentation, which can be time-consuming and exhausting.

The Waterfall model worked well in the past when the applications were relatively less complex; thus, the stakeholders had clear requirements upfront and the large organizations would then dedicate fixed budgets and resources to deliver the project. However, these methods fundamentally lack flexibility and collaboration in the development process whereby any changes in the requirement, error in the design, or a problem discovered during testing can be disastrous forcing a major step backward in the process. The "silos" effect often prevents the team from establishing the necessary cohesion required for faster delivery techniques like **Continuous Integration (CI**) and **Behaviour Driven Development (BDD**). Also, as the phase is sequential any delays in the previous phases can cascade into the subsequent phases, delaying and slowing the overall software development cycle.

Owing to all the challenges introduced by sequential methods, organizations started transitioning to other methods which provide more agility, visibility, collaboration, and flexibility. Today organizations and startups are working on solving some complex real-world problems with ever-changing requirements with limited funding and resources. Thus, it becomes paramount for any modern-day organization to have agility, flexibility, and visibility in its software development life cycle.

In today's modern world, many organizations are adopting agile methodologies, which focus on incremental and iterative development, where requirements and solutions are created through collaboration between self-organizing cross-functional teams. Agile development allows teams to deliver value faster, with more predictability and quality, and to respond more effectively to changes. Agile methodologies, such as Scrum and Kanban, have revolutionized software development by introducing principles and practices that prioritize flexibility, collaboration, iterative development, and faster time-to-market. These methodologies promote adaptability in response to changing requirements and market dynamics. By embracing flexibility, teams can adjust their plans and priorities as needed, ensuring that the final product aligns with customer needs and expectations. Collaboration is at the core of agile methodologies, as they encourage cross-functional teamwork and effective communication. By fostering collaboration between individuals with different expertise, such as developers, testers, and designers, agile methodologies enable a collective and shared understanding of the project. This collaborative environment promotes knowledge sharing, problem-solving, and continuous learning, resulting in higher quality solutions. Because of the benefits, adoption of Agile methodologies has been increasing as shown in the diagram:



Waterfall methodology
Search term

Agile methodology
Search term

Worldwide ▾ 2004 - present ▾ All categories ▾ Web Search ▾

*Figure 1.2: Google trend showing the growing popularity of Agile vs Waterfall*

Agile can address most of the shortcomings of Waterfall method by inherently addressing the issues mentioned as follows:

- **Project Scope**: Agile works well even if the scope is not well-defined. It is easier to make changes to the project after it starts but may incur expenses.
- **Timeline**: Agile has no fixed timeline. This is better for longer or continuous projects that require greater innovation.
- **Budget**: Agile allows the team to have a flexible budget. This increases funding efficiency.
- **Flexibility**: Agile is very flexible. Allows for innovation and collaboration and makes it easy to change project courses.
- **Customer Involvement**: In Agile, customers are involved in every phase of **Software Development Life Cycle (SDLC)**.
- **Risk**: Agile carries less risk because the product is frequently tested throughout the project.

# Agile is Good, so why do we need DevOps

Agile and DevOps are two distinct methodologies that complement each other in the software development process. Agile methodology is focused on rapid iteration and continuous delivery, while DevOps is focused on streamlining the software development and deployment process.

Agile methodology helps software development teams to deliver high-quality software quickly by breaking down complex projects into smaller, manageable tasks. It promotes continuous delivery, customer feedback, and collaboration among team members.