

DevOps Design Patterns

*Implementing DevOps best practices for
secure and reliable CI/CD pipeline*

Pradeep Chintale



www.bpbonline.com

Copyright © 2024 BPB Online

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

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

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

First published: 2024

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55519-924

www.bpbonline.com

Dedicated to

My beloved Parents:

Shankar Chintale

Kamal Chintale

&

*My wife **Prachi**, my daughter **Prisha** and son **Pransh***

About the Author

Pradeep Chintale is a Lead Site Reliability Engineer at SEI Investment Company with over 17 years of extensive experience. He has completed his bachelor's degree in Computer Science at Mumbai University. Throughout his career, he has served as a System Analyst, specializing in Infrastructure automation, Cloud and DevOps Architecture, and providing support for Application and Enterprise Infrastructure Management.

His primary expertise lies in Cloud and DevOps architecture, where he excels in designing and implementing automated and continuous build, deploy, and release processes using integrated tools within the CI/CD pipeline. He is proud to hold certification as a Cloud Solution Professional Architect.

In his current role at SEI, he continues to contribute to the field as a Lead Site Reliability Engineer. Previously, He served as a Senior Cloud Solution Architect at Microsoft, where his responsibilities included transforming technical project requirements into comprehensive architecture and design solutions. His focus has always been on establishing robust, highly available, and scalable CI/CD platforms for the fully automated deployment and operation of workloads.

When it comes to expertise in DevOps and Cloud technologies, He is recognized as a top professional in the field.

About the Reviewers

- ❖ **Nikhil Kumar** is a seasoned software engineer based in India, boasting over six years of industry experience. With a versatile skill set spanning development, architecture, and consultancy, Nikhil has worked across diverse domains for various clients and employers.

His expertise includes DevOps, Ansible, Python, CI/CD, Kubernetes, and Shell Scripting. Nikhil holds multiple prestigious certifications from RedHat, including Red Hat Certified Engineer (EX294) and Red Hat Certified System Administrator (RHCSA). As a prolific technical blogger, Nikhil shares his wealth of knowledge with the broader software engineering community.

- ❖ **Anupam Singh**, SRE practitioner leading a major project for a financial technology organization. She has worked as DevOps and SRE across various domain industries and successfully delivered solutions. She is an experienced software developer with deep understanding of SLDC from different facets. Author to various articles on Medium.com. Passionate about technology and solving people's lives through technology. She has a master's degree in MIS from Arizona State University and a Bachelors of Technology in Information Technology from GBPUAT Pantnagar.

Presently working with American Express as SRE Director for the last two years and is responsible for global payments.

Outside work, she also volunteers for a global nonprofit organization, AnitaB.Org India Delhi Chapter and organized events for women in technology.

On personal front, she loves travelling. In her free time, she is mostly with her garden plants and spending time gardening. She is also a health enthusiast and loves to explore new adventures and activities.

Acknowledgement

There are a few people I want to thank for the continued and ongoing support they have given me during the writing of this book. First and foremost, I would like to thank my wife, Prachi, for continuously encouraging me to write the book, I could have never completed this book without her support.

I am grateful to the course and the companies which gave me support throughout the learning process of DevOps Design Pattern and as it is very important to learn the tools related to DevOps Design Pattern.

Thank you for the all hidden support provided. I gratefully acknowledge Mr. Vibhu Bansal for his kind technical scrutiny of this book.

My gratitude also goes to the team at BPB Publication for being supportive enough to provide me quite a long time to finish the first part of the book.

Preface

This book covers many different aspects of **DevOps Design Patterns**, and the importance of DevOps best practices. This book also introduces the importance of DevOps Design Patterns in the field of real-time industry. It shows how the DevOps is important for the industries. This book is intended for individuals who already possess a fundamental understanding of DevOps but aspire to advance their knowledge and acquire expertise in DevOps best practices. It also gives importance to Software automation, **Infrastructure as a Code (IaC)**, **continuous integration (CI)** and **continuous deployment (CD)**.

In this book, you will learn about DevOps best practices. An important aspect of DevOps includes the implementation of CI and CD.

This book will cover a deep dive into DevOps architecture, design, and implementation of continuous build, continuous deploy, and continuous testing processes using integrated tools in the CI/CD pipeline like Github, Jenkins, Kubernetes, and AWS/ Azure Cloud.

This book will provide an interdisciplinary skill set to cultivate a continuous deployment capability in your organization. The reader will be an expert in implementing DevOps best practices.

After completing this course, you will be able to:

- Explain the skill sets and roles involved in DevOps and how they contribute toward a continuous delivery capability.
- Review and deliver automation tests across the development stack.
- Explain the key jobs of system operations and how today's leading techniques and tools apply to them.
- Explain how high-functioning teams use DevOps and related methods to reach a continuous delivery capability.
- Facilitate prioritized, iterative team progress on improving a delivery pipeline

This book is divided into 11 chapters. After reading this book, readers are going to have competitive knowledge of DevOps design, architecture, and its pros and cons. This book has complete guidelines on how we can implement a DevOps best practice.

Chapter 1: Why DevOps – This chapter delves into the fundamentals of DevOps for the sake of comprehension. It explores the evolution of DevOps development and elucidates

how it contributes to achieving cutting-edge technology, shortening the life cycle of system development, streamlining processes, enhancing business efficiency, lowering operational costs, and maintaining a competitive market position.

Chapter 2: Implement Version Control and Tracking – This chapter provides an in-depth exploration of version control, also known as source control. Teams contribute changes in the form of revisions, allowing the seamless merging of work at precise moments. The diverse operational possibilities within version control systems empower teams to choose methods aligned with their utilization of branching and merging features.

This section provides a comprehensive and detailed explanation of version control, outlining its benefits and the mechanisms for exercising control over it.

Chapter 3: Dynamic Developer Environment – In this study, we’re exploring ways to make things easier for developers. We’re focusing on flexible and on-demand environments. Software companies are interested in technologies that can be easily adjusted and expanded to make deploying software faster and more efficient. When developers are working on a new feature, it’s really helpful to have a safe space to test and develop it without causing issues for the rest of the team. This way, we can be more confident about making changes and have a spot to test them before adding them to the main part of the project.

Chapter 4: Build Once, Deploy Many – In this part of the book, we will talk more about the Build Once, Deploy Many way of doing things. We will look at why it is good, what problems it might have, and the best ways to use it. This information is for developers, project managers, and others who work on making software. We’ll give a complete guide with the tools and tricks you need to use this approach well. By the end of the book, reader will have a thorough understanding of the Build Once, Deploy Many strategy and they will be ready to use it in their own software projects.

Chapter 5: Frequently Merge Code: Continuous Integration – In this chapter, we are going to cover continuous integration (CI), which is a software development practice that involves regularly and automatically building, testing, and integrating changes made to a codebase. The goal of CI is to catch and fix problems in the codebase as early as possible in the development process to prevent those problems from causing larger issues down the line.

Chapter 6: Software Packaging and Continuous Delivery – In this chapter, we will take a deeper look into continuous delivery (CD), and describe how this phase of the process is the key to achieving greater efficiency in your software development life cycle.

CD may sound daunting to teams already stretched to the limit. But once established, these game-changing practices and the automation that comes with them can take your software delivery practices to the next level. These deliveries should be frequent, carrying incremental changes to the code, which makes releases low-risk, low-stress events for DevOps teams and seamless for end-users with little or no downtime.

Chapter 7: Automated Testing – In this chapter, we will learn about how to improve the overall quality and efficiency of the software development process by automating repetitive and time-consuming testing tasks, along with minimizing the impact of issues and improving the traceability of tests by integrating automated tests into the CI/CD pipeline and monitoring the system’s performance over time.

Chapter 8: Rapid Detection of Compliance Issues and Security Risks – In this section, we will learn how to improve the overall quality and efficiency of the software development process by automating repetitive and time-consuming testing tasks. To minimize the impact of issues and improve the traceability of tests by integrating automated tests into the CI/CD pipeline and monitoring the system’s performance over time.

Chapter 9: Rollback Strategy – In this chapter, we will learn the importance of having a DevOps rollback strategy to minimize the impact of issues arising during or after deployment. We will delve into the key components of a successful rollback plan, including automated rollback processes, version control, testing, communication plans, and post-rollback analysis. Additionally, we will provide the best practices and tips for implementing a DevOps rollback strategy that is efficient, reliable, and minimizes downtime.

Chapter 10: Automated Infrastructure – In this chapter, we will learn how to improve the overall quality and efficiency of the software development process by automating repetitive and time-consuming testing tasks. We will also delve into how to minimize the impact of issues and improve the traceability of tests by integrating automated tests into the CI/CD pipeline and monitoring the system’s performance over time.

Chapter 11: Focus on Security: DevSecOps – In this chapter, we will learn about DevSecOps. It is an approach to software development that integrates security practices into the Development Operations (DevOps) process. It emphasizes the importance of security early in the software development lifecycle rather than treating it as an afterthought.

Code Bundle and Coloured Images

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

<https://rebrand.ly/neoyh4f>

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

<https://github.com/bpbpublications/DevOps-Design-Patterns>.

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. Why DevOps.....	1
Introduction.....	1
Structure.....	1
Objectives.....	2
What is DevOps.....	2
<i>Classification.....</i>	2
<i>Usage of DevOps in the modern world</i>	3
<i>Advantages of DevOps</i>	4
<i>Disadvantages of DevOps</i>	5
Agile and DevOps go hand-in-hand	6
DevOps best practices.....	8
<i>Involvement of stakeholders in the method.....</i>	9
<i>Automated testing and building environments.....</i>	9
<i>Integrated configuration and change management</i>	9
<i>Continuous integration and continuous deployment.....</i>	9
<i>Continuous delivery and product support.....</i>	10
<i>Application monitoring and automation of dashboards</i>	10
Best tools to use for CI/ CD pipeline	10
Jenkins.....	11
TeamCity.....	11
CircleCI.....	12
Travis CI	12
GitLab CI	13
Bamboo.....	13
Semaphore.....	14
GoCD.....	15
Azure DevOps	15
CodeShip	16
Bitrise.....	16
Drone CI	17
Challenges that DevOps helps overcome	17

Overcoming the dev versus ops mentality.....	18
Common understanding of CD practices	18
Moving from legacy infrastructure and architecture to microservices	18
Implementing a test automation strategy.....	18
Too much focus on tools	19
Team ownership for deployments and releases.....	19
Resistance to change	19
Key metrics are being acted upon	20
Dev and Ops toolset clashes	20
Getting started with continuous learning.....	20
Measuring the success of DevOps implementation	21
Evaluate the need to implement DevOps practice.....	21
Break the organizational silos and encourage collaboration	21
Put customer/end-user satisfaction at the center	21
Do not jump start, instead, start small and then scale up	22
Automate wherever possible	22
Select tools that are compatible with each other	22
Define performance reviews for team and an individual	22
Ensure real-time visibility into the project	23
Integrate and deliver continuously.....	23
Achieve better results with monitoring and feedback.....	23
Conclusion.....	24
2. Implement Version Control and Tracking	25
Introduction.....	25
Structure.....	25
Objectives.....	26
What is Version Control or Source Control	26
What is source code management	26
Benefits of using Version Control.....	26
Version Control tools in DevOps.....	27
GitHub.....	27
GitLab	27
Bitbucket	28
Perforce	28
Apache subversion	28

Hg	28
How does version control work.....	29
Why is Version Control necessary	30
Simpler and better bug suppressing.....	30
Concurrent growth.....	31
Greater dependability of the final result	31
Decompose by business capability	32
Problem: How to decompose an application into services	33
Problems: How do you establish a business capability.....	34
Guidelines for using Git.....	34
Create clear, focused commitments.....	34
Commit early, commit often	35
Conclusion.....	36
3. Dynamic Developer Environment.....	37
Introduction.....	37
Structure.....	37
Objectives.....	38
Why feature environment	38
Accelerating the installation process	42
Information processes automation and orchestration.....	42
Coordination of operations and production organizations	43
What is agile in software development.....	43
Agile methodologies.....	44
Last impressions	45
Faster delivery.....	47
Creating a shorter route between points a and b	47
Send out little medals	48
What are our pain points.....	48
Ideal solution.....	49
Atmosphere for entrepreneurship shortcomings.....	50
Infrastructure changes	50
Routing of feature environments	51
Robotics to the rescue.....	51
Fundamental concept is a three-step procedure.....	51
Accelerated shipping.....	52

<i>Our machining might differ</i>	52
Automate deployment to feature environments	53
<i>What exactly is a features atmosphere</i>	53
<i>What is the point of creating features climates</i>	53
<i>Presence of characteristic elements in all habitats</i>	53
Removal of obsolete feature environments	54
Conclusion.....	55
4. Build Once, Deploy Many	57
Introduction.....	57
Structure.....	58
Objectives.....	58
Build Once Deploy Many	58
<i>Library dependencies</i>	60
<i>Testing limitations</i>	61
<i>Wrapping up</i>	61
Building docker image.....	62
<i>Docker image use cases</i>	63
<i>Docker container vs. Docker image</i>	63
<i>Anatomy of a Docker image</i>	64
<i>Docker image repositories</i>	65
<i>Creating a Docker image</i>	65
<i>Interactive method</i>	66
<i>Docker file method</i>	66
<i>Docker image commands</i>	66
Environment overlays.....	67
<i>How do overlays affect application deployment</i>	69
Continuous deployment.....	70
<i>Continuous deployment vs. continuous delivery</i>	70
<i>Continuous deployment tools</i>	71
Pros and cons	72
Conclusion.....	73
5. Frequently Merge Code: Continuous Integration	75
Introduction.....	75
Structure.....	76

Objectives.....	76
Traditional software development.....	76
Commit early, commit often	82
Build only once	85
Clean your environments.....	88
Monitor and measure your pipeline.....	92
Keep the builds green	94
Streamline your tests.....	97
Make it the only way to deploy to production.....	99
Make it a team effort	102
Best tools to build robust CI process	106
Step-by-step building robust and fully automatic CI process	111
Conclusion.....	115
6. Software Packaging and Continuous Delivery.....	117
Introduction.....	117
Structure.....	118
Exploring the benefits of CD.....	119
<i>Increasing developer productivity</i>	<i>121</i>
<i>Simplifying implementation through automation.....</i>	<i>123</i>
<i>Accelerating feedback delivery.....</i>	<i>123</i>
<i>Enhancing testing quality</i>	<i>124</i>
<i>Expediting market introduction of new capabilities</i>	<i>125</i>
<i>Selecting the tools for a robust CD process</i>	<i>125</i>
AWS Code Deploy.....	126
Octopus deploy.....	126
Jenkins	126
TeamCity	127
Deploy Bot.....	127
GitLab.....	127
Bamboo	127
Circle	127
Code ship	127
Google Cloud Deployment Manage	127
<i>Building a robust and fully automatic CD process</i>	<i>128</i>
Measuring considerations for calibrating a CD pipeline.....	128

<i>Analyzing the lead time in a CD pipeline</i>	129
<i>Evaluating the cycle time in a CD pipeline</i>	130
<i>Assessing the Mean Time to Recovery</i>	131
<i>Examining defect resolution time</i>	132
<i>Understanding the test pass rate</i>	133
Best practices for adopting continuous delivery	135
<i>Develop Service Level Objectives</i>	136
<i>Automating SLO evaluation with quality gates</i>	137
<i>Automating every redundant process.....</i>	137
<i>Keeping everything in version control.....</i>	138
<i>Providing fast, useful feedback</i>	140
<i>Deploying the same way to every environment.....</i>	141
<i>Avoiding direct changes in the production environment</i>	142
<i>Deploying to every environment the same way.....</i>	143
<i>Deploying in a copy of production.....</i>	143
<i>Including the database</i>	144
<i>Eliminating complexity</i>	145
<i>Establishing observability and continuous monitoring</i>	146
<i>Conclusion.....</i>	146
7. Automated Testing.....	147
Introduction.....	147
Structure.....	147
Objectives.....	148
Adopting test automation	148
Introducing tool integration.....	150
Tracking metrics.....	152
Leverage containerization	154
Keeping communication transparent	156
Saving time with headless execution.....	158
Exploring multi-layer tests.....	161
Integrating performance testing into delivery cycle	163
Building robust continuous testing with tools.....	167
Continuous testing	168
<i>How test management tools work.....</i>	171
Step-by-step build robust and fully automatic continuous testing.....	172

Conclusion.....	177
8. Rapid Detection of Compliance Issues and Security Risks	179
Introduction.....	179
Structure.....	180
Objectives.....	181
Types of continuous monitoring	181
Network monitoring	182
<i>Why network monitoring</i>	<i>183</i>
<i>Benefits of network monitoring</i>	<i>184</i>
<i>How to use network monitoring</i>	<i>185</i>
<i>Configuring network monitoring</i>	<i>185</i>
Infrastructure monitoring	186
<i>Benefits of infrastructure monitoring.....</i>	<i>187</i>
Configuring infrastructure monitoring.....	188
Application Performance Monitoring.....	190
<i>Benefits of Application Performance Monitoring.....</i>	<i>190</i>
<i>When to use Application Performance Monitoring</i>	<i>191</i>
<i>Web application monitoring on Azure</i>	<i>192</i>
<i>Application Insights</i>	<i>197</i>
<i>Azure Container Insights</i>	<i>199</i>
<i>Azure VM Insights.....</i>	<i>200</i>
Setup alerts	201
<i>Visualize monitoring data.....</i>	<i>203</i>
<i>Dashboards</i>	<i>204</i>
<i>Azure workbooks.....</i>	<i>204</i>
<i>Power BI</i>	<i>205</i>
<i>Configuring automatic continuous monitoring.....</i>	<i>206</i>
Conclusion.....	207
9. Rollback Strategy	209
Introduction.....	209
Structure.....	209
Objectives.....	210
Introducing rollback strategies.....	210
<i>Manual rollback procedures.....</i>	<i>212</i>

Automated rollback scripts	213
Snapshot backups.....	213
Version control.....	217
Hotfix rollback strategy	221
Feature toggles rollback strategy	223
Immutable infrastructure rollback strategy.....	225
Canary deployment.....	226
A/B testing deployment strategy	229
Blue-green deployment	231
Shadow deployment	233
Conclusion.....	234
10. Automated Infrastructure.....	237
Introduction.....	237
Structure.....	237
Objectives.....	238
Infrastructure as Code	238
How to choose the best IaC tool	240
Introducing Terraform	241
Pipeline as a Code	244
Platform as Code	247
Configuration as Code.....	251
Policy as Code.....	253
GitOps methodology	257
GitOps WorkFlow.....	258
Use cases of GitOps	260
Best tools for IaC.....	261
Conclusion.....	263
11. Focus on Security: DevSecOps.....	265
Introduction.....	265
Structure.....	266
Objectives.....	266
Principles of DevSecOps.....	267
Automate security processes.....	269
Collaborate across teams	270

Implementing security by design	271
<i>Using secure coding practices</i>	272
Integrating compliance and governance.....	273
Implementing DevSecOps training and education	274
Integrating security into the deployment pipeline.....	274
Maintaining visibility and control	275
Collaboration between DevOps and security	276
<i>Threat modelling</i>	277
<i>Compliance as code</i>	278
<i>Container security</i>	279
<i>DevSecOps metrics</i>	280
<i>Securing the public endpoints</i>	281
<i>Define policy and governance</i>	282
<i>User right network tools to filter traffic</i>	282
Define and implement IAM, RBAC and 2FA	283
Implementing least privilege model.....	284
Segregating DevOps network.....	285
Using password manager.....	286
Conclusion.....	287
Index	289-298

CHAPTER 1

Why DevOps

Introduction

In this chapter, the basics of DevOps are discussed for understanding purposes. The background behind the DevOps development is addressed. How this delivers the highest technology, shortens the system development life cycle, optimizes processes, improves business time, minimizes operating expenses, and retains market position is explained in this chapter.

Structure

In this chapter, we will learn:

- What is DevOps?
- Agile and DevOps go hand-in-hand
- DevOps best practices
- Best tools to use for CI/CD pipeline
- Challenges that DevOps helps overcome
- How do we measure the success of DevOps implementation?

Objectives

In this chapter, we will look into what DevOps is and what are the advantages and disadvantages of DevOps. DevOps best practices in terms of execution and implementation. We are going to cover what are the best suitable tools for making a robust CICD pipeline in this book based on real time scenarios.

What is DevOps

DevOps combines **development (Dev)** and **operations (Ops)** to integrate people, processes, and technologies in application design, development, and operations. Development, IT operations, quality engineering, and security can now coordinate and collaborate thanks to DevOps.

Organizations use DevOps culture, methods, and technologies to build quality applications on time, improve their responsiveness to user requests, and accelerate the achievement of business objectives. DevOps enables teams to continuously give value to consumers by creating more dependable and superior products.

Classification

Academicians and practitioners have not come up with a common meaning for the term *DevOps* other than that it is a cross-functional combining (and a portmanteau) of terms and notions for *developing* and *procedures*. Shared ownership, transaction processing, and efficient communication are some of the most common characteristics of DevOps. Software development and (telecom) operational processes were brought together in 1993 by the Telecommunication services based on organizational infrastructure collaboration. The inaugural DevOps days conference took place in *Ghent, Belgium*, in 2009. Belgian professional, project coordinator and agile operator *Patrick Debois* started the convention. Other countries have joined the meeting.

DevOps is a set of methods that enable enterprises to speed up innovations, deliver the highest technology, shorten the system development lifecycle, optimize processes, improve business time, minimize operating expenses, and retain market position. Automated provisioning, automation testing, automation build and deployment, and continuous evaluation are all part of the DevOps system. Development and IT operations are brought together in the term *DevOps (Ops)*. It seeks to reduce the time it takes to design a system and produce high-quality software on a regular basis. Agile software development and DevOps go hand in hand, and numerous components of DevOps are derived from Agile.

When it comes to the management of an organization's products, DevOps is critical. DevOps supports companies in a wide range of industries, from start-ups to large multinationals, by providing everything from business strategy to operational oversight.

In the following *Figure 1.1*, the various steps for DevOps are presented:

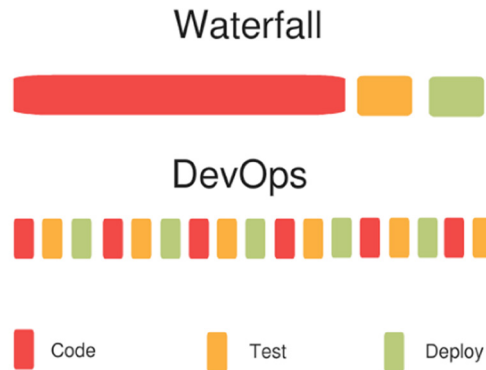


Figure 1.1: Traditional (Waterfall) s/w model vs DevOps model

The following figure shows the steps of DevOps:

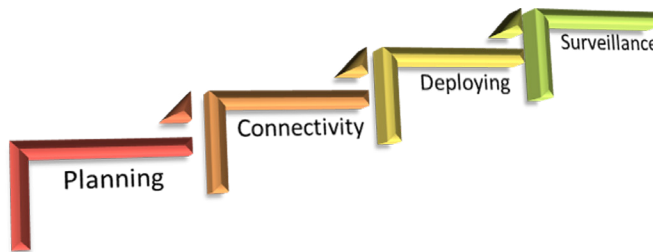


Figure 1.2: Steps for DevOps

- **Planning:** Organizing work into smaller cycles is an important first step in implementing DevOps in your firm.
- **Connectivity:** The new functions are integrated into the current programming language during this step. System testing and constant improvement are essentially the same thing.
- **Deploying:** At this point, the implementation process is ongoing. Developers will be able to make changes to the software without affecting its performance.
- **Surveillance:** As part of this phase, the production team must maintain a close eye on structural response and look for bugs.

Usage of DevOps in the modern world

Before the adoption of DevOps, teams operated in isolated units, lacking communication and collaboration between IT operations and developers. It was common for programmers to hand off their programming to the operator, who was responsible for maintaining it. Supervisors and programmers both lacked sufficient knowledge of operating processes and software packages. Consequently, as software developers prioritized the efficient release of new features, system administrators placed a higher emphasis on ensuring the

long-term reliability of the system. This divergence led to miscommunication, delayed software deployments, and increased operational costs.

DevOps emerged to address the issues of software development, testing, deployment and delivery. It is a combination of strategies to remove the obstacles that inhibit designers, administrators, and other company segments from functioning together more effectively and profitably. In DevOps, the creation and deployment teams can work together more effectively because of the integration of the two groups. Regular testing of app efficiency and durability, together with automation of architectures, might help achieve these goals.

Once we understand DevOps as a Service, let us look into DevOps as a Service in more detail. At its most basic level, DaaS is a delivery model that is severe enough to require the common use of a toolkit to maintain records of all their operations on the cloud infrastructure. When you use DaaS, you get the possibility of automating the distribution of your project's technology. DevOps engineers are in charge of overseeing the whole product development process from start to finish, making sure that nothing goes wrong along the way. For your projects, they know which technologies and tools work best and how to use them to their full potential.

Advantages of DevOps

The use of cloud services has the capability to assist a business in developing tremendously. In the following *Figure 1.3*, the advantages of DevOps are presented. Let us have a look at the benefits of DevOps in the corporate world:



Figure 1.3: Advantages of DevOps

- **Improves Business Acumen:** In an organization, corporate success depends on its ability to adapt quickly to changing conditions. Businesses can reach new heights of success thanks to DevOps. As everything is more data-driven with DevOps services, everyone on the team is using the same information. By using these technologies, businesses are able to organize and control their operations in a far more comprehensive manner.
- **Workplace that encourages teamwork:** Using the DevOps method, you may create a more stable working environment. When using a cloud-based DevOps platform, communication is significantly easier because all tools deployed in the virtualized environment can be accessed from almost anywhere.
- **Quality enhancement:** With DevOps, difficulties may be found and fixed much more quickly than with the previous way of doing things. Automated build, deployment and problem solving help early identification of issues that help deliver the product on time. The quality of a product can be greatly improved by working together efficiently between the developmental and operational teams and collecting feedback from customers on a regular basis. DevOps as a Service enables rapid testing and deployment of fresh features. The regularity of releases tends to increase when cloud services are used. Developers have access to additional computing power and data capacity.
- **Enhancing efficiency by simplifying operational procedures:** This method simplifies the understanding of information and data movement, but it can also result in team members lacking awareness of the holistic operation of the complete toolchain. IT managers can use managed services tools to make changes, while software developers can utilize source code administration solutions to test their work. DevOps experts are immediately available to construct projects using DaaS, allowing them to go to work on your assignment right away.

Disadvantages of DevOps

Testing of operational processes in the cloud's manufacturing process is more difficult and raises questions about connectivity. Interoperability, technology, and workflow coordination require a deep understanding of the subject matter. In DevOps, efficiency takes precedence over security.

- A defect can be found in any technology or system, and there is no one-size-fits-all solution to our problems. Even though it is an excellent idea, DevOps as a Service is far from perfect.
- Companies will need specialized personnel to help them implement and operate new approaches during this stage.
- A detailed understanding of DaaS products' interaction with the business framework, infrastructure, and operating methods is essential.
- DevOps places an emphasis on speed over security while developing software.