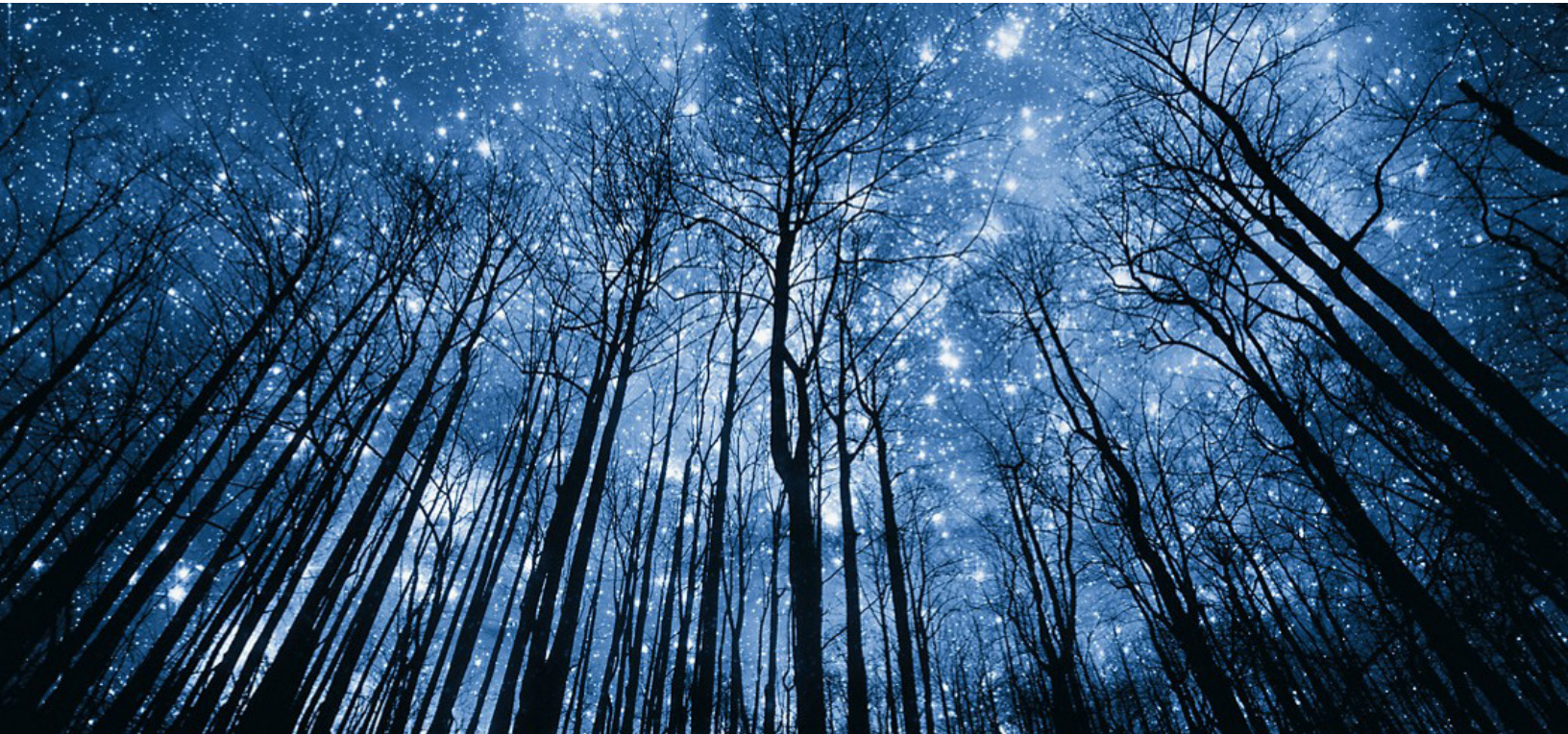


A BEGINNER'S HANDBOOK TO DEVOPS



Aashikha Narasimhan

Student, Master of Engineering Management
Duke University

Roshan Sham

Solutions Engineer
JumpCloud

Harish Subramanian

Inside Sales Engineer
Commvault

The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across multiple technologies and products.

From Associate, entry-level courses to Expert-level, experience-based exams, all professionals in or looking to begin a career in IT benefit from industry-leading training and certification paths from one of the world's most trusted technology partners.

Proven Professional certifications include:

- Cloud
- Converged and Hyperconverged Infrastructure
- Data Protection
- Data Science
- Networking
- Security
- Servers
- Storage

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

Learn more at www.dell.com/certification

Table of Contents

| | |
|--|----|
| ABSTRACT..... | 4 |
| WHAT IS DEVOPS..... | 5 |
| HISTORY OF DEVOPS..... | 6 |
| TIMELINE OF DEVOPS..... | 8 |
| LIFECYCLE OF DEVOPS..... | 9 |
| BENEFITS OF DEVOPS..... | 11 |
| INDUSTRY PRACTICES USING AZURE DEVOPS..... | 12 |
| WHAT IS DEVSECOPS?..... | 14 |
| PRESENT-DAY UTILIZATION PROS AND CONS..... | 15 |
| FUTURE SCOPE OF DEVOPS..... | 17 |
| CONCLUSION..... | 18 |
| REFERENCES..... | 19 |

ABSTRACT

Application development in a fast-paced world of technology relies heavily on people, processes, and technology to deliver reliable software quickly. Initially heavily dependent upon the waterfall model, a top-down approach encompassing gathering requirements, design, implementation, verification, and maintenance simplified the development lifecycle but had some areas for improvement, including a lack of visibility of the project and difficulty catering to changing requirements, among others. As an improvement, the agile methodology split the waterfall model into different sprints, performed multiple iterations, released phases of an application with critical features, and introduced advanced features with new releases to achieve desired outcomes. DevOps, however, gained popularity as IT operations and software development teams grappled with the time it took to deliver an application using traditional models. DevOps metamorphosed agile application development by enabling a single unit to perform continuous development, integration, testing, deployment, and monitoring.

This paper will explore the history of software development models and the coming to existence of DevOps. It would also dive into the different processes associated with each phase of the DevOps lifecycle and explore DevSecOps on a high level. The whitepaper is intended for readers looking to gain insight into application development processes and fully appreciate how DevOps changed the world.

WHAT IS DEVOPS

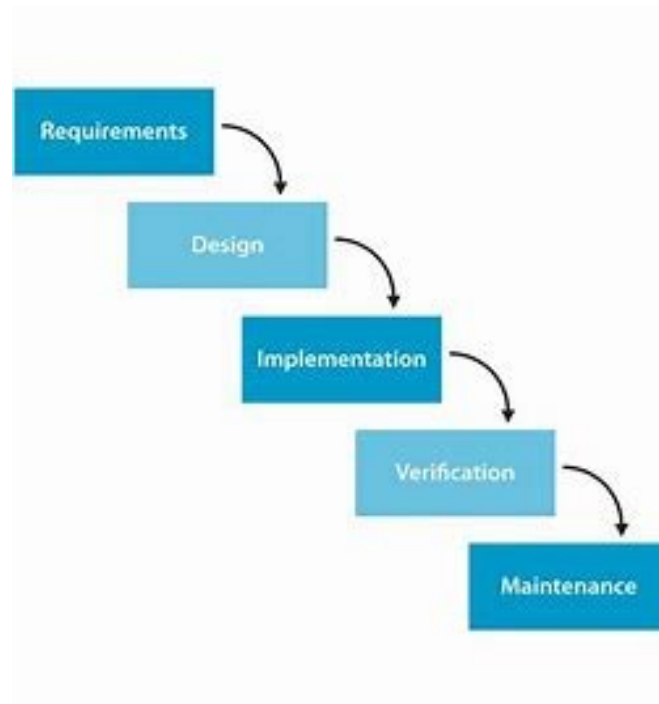
Software development in the age of the Internet has transformed the world across all industry segments, from IT to banking, marketing, and entertainment. Companies drive their annual and long-term targets with sales statistics and continued interaction with their customer base via software, applications, and devices. In addition to using customer input to drive strategy, they leverage tools to increase operational efficiencies to drive their logistics and supply chain. Businesses realized, albeit quickly, that they needed to build and update their software just as soon as they delivered products via industrial automation to become a market leader in their respective segments.

DevOps, essentially an abbreviation of two words- development and operations, is a methodology to manage and monitor the complete lifecycle of application development and deployment by enabling a single team. DevOps involves continuous integration, deployment, monitoring, communication, and collaboration to automate and accelerate processes. Software developers and operations personnel collaborate to eliminate silos in the software development lifecycle to work on and release new features, updates, and fixes in synchronization with business objectives. The end goal of DevOps is to deliver software of superior quality and reliability in the shortest period.

HISTORY OF DEVOPS

To understand why DevOps came into existence, let us look at the success story of Netflix. This media platform functions and rolls out new movies daily, requiring constant updates. As a modern customer, it is hard to imagine a world where Netflix only deploys once a month or every few months. This was the unfortunate reality for developers back in the day.

So, what exactly is the waterfall model? It was one of the earliest known methodologies used in the field of software development, which usually includes a 7-step process, i.e., Requirement, Design, Implementation, Verification, and Maintenance. So, let's relate this to the Netflix example, where the software would have first been developed and thoroughly tested for months before its market release. The Quality Assurance (QA) team would have had to take guesses to test every possible scenario and return defects. In contrast, developers would have had to hustle on bug-fixing and sending them faster to QA so that they could run them through tests. Due to the number of people involved and the complexity, teams often had to ask the management to postpone the software's release date.



1

¹ Water-Fall Model Design [Refer 1]

The stakeholders who are involved would present an entirely different approach. It would often push the team to the edge, leaving them no choice but to roll back everything or release a pre-mature version of the software, followed by support releases for restarts, hotfixes, etc., to reduce the impact of the damage.

However, this is only a temporary solution because the team can stabilize the software only until the next release. This ended up being a cyclic process of releasing inefficient software.

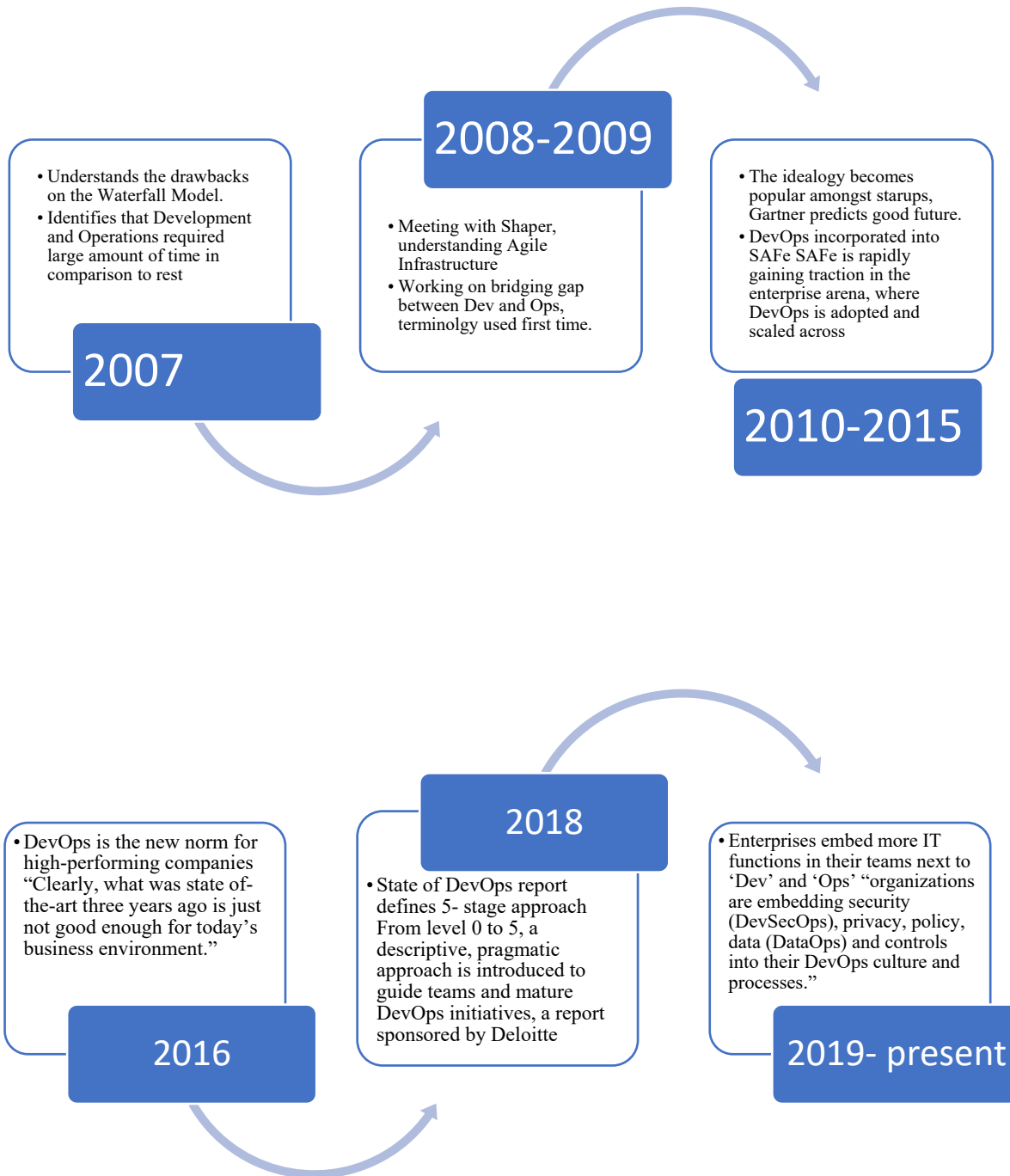
Following the failures and complexities set by the waterfall model, a newer methodology was introduced called the Agile approach, an iterative approach to project management and development that helps teams develop software at a faster, more efficient way, with lesser rollbacks. This methodology delivers work in small but regular increments. The different phases are analyzed continuously, hence the developers have a natural mechanism to change their approach quickly. However, the agile methodology was not the best in the market and came with its own shortcomings. Since the project was divided into multiple phases, it led to resource planning constraints and documentation fallacies, which resulted in fragmented output without a finite end, making it difficult to measure.

A newer methodology called DevOps was formally termed and created in 2009 by Patrick Debois. Debois is now regarded as one of the pioneering figures of DevOps and has gained significance over the years as one of its gurus, with more and more organizations integrating DevOps into their operating systems.

The evolution of DevOps occurred because of the combination of the words, ‘development’ and ‘operations,’ which provides a fundamental point for comprehending what exactly people mean when they refer to “DevOps.” The DevOps methodology is not a technology, process, or an established definitive.

DevOps is often referred to as a cultural viewpoint - the true meaning of DevOps has widened to become an umbrella term referring to the culture, processes, and mindset used for optimizing and shortening the life cycle of software development with the help of fast feedback loops for offering features, updates, and fixes at a frequent pace.

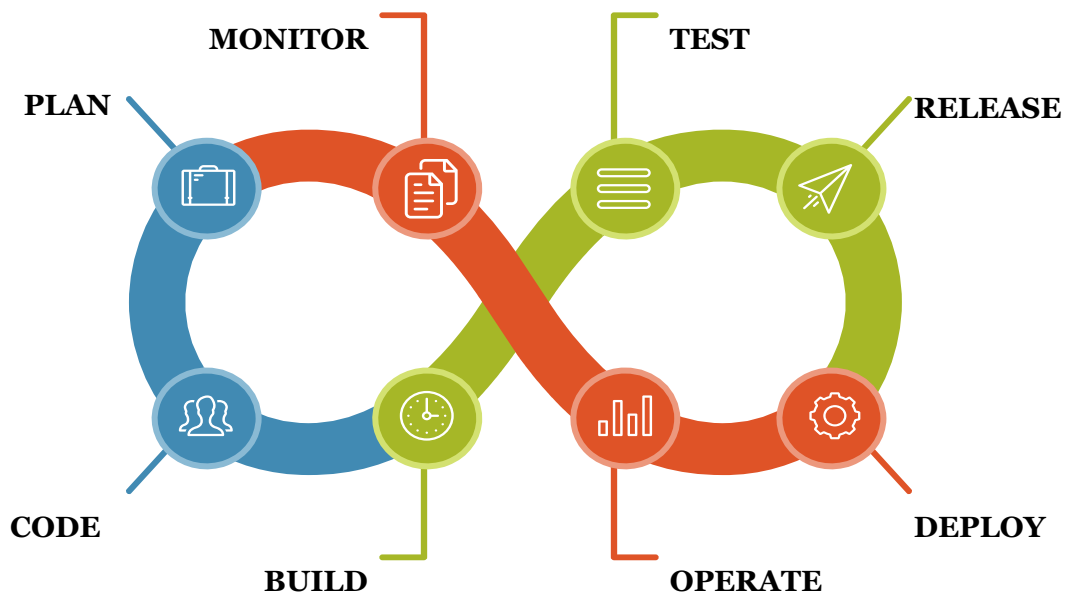
TIMELINE OF DEVOPS



LIFECYCLE OF DEVOPS

Now that we understand the history of software development and why we needed a more efficient development methodology, let us know the DevOps Lifecycle.

Plan: In DevOps, planning plays a vital function; all the requirements of the project, which include the challenges like time taken for every stage, price, and many other parameters, are discussed, and this can help everyone working on the project teams get an idea of the project.



Code: In this stage, the code is written in line with the client's requirements. The code is split into smaller fragments called Units, which help get a clear picture of the code. For example, on a medical billing system, the login code can be considered as one unit, and billing categories can be regarded as another unit.

Tools- JIRA, Git

Build: The units (small codes) are built at this stage.

Tools- Gradle, Maven

Test: The team performs tests across all the units to ensure no bugs. If found, it will be returned.

Tools- Pytest, Selenium

Integrate: In this stage, all the units of the codes are integrated, which helps build a connection between the development team and operation team to put into effect Continuous Integration and Continuous Deployment (CICD).

Tool- Jenkins

Deploy: The code would be deployed on the client's environment.

Tools- AWS, Docker

Operate: Any operations on the code are performed in this phase.

Tool- Kubernetes

Monitor: As the application is mounted on the client's environment, the application behavior and the client environment are monitored across different operations making sure of the best application performance.

Tool- Maven

TOOLS USED IN THE DEVOPS LIFECYCLE: A BRIEF REVIEW

Git is an open-source distributed version management software designed to handle many small to substantial projects with pace and performance. Git has the functionality, overall performance, protection, and versatility that most teams and coders need.

Gradle is an automation tool known for its flexibility in building software and automating the creation of applications. The building process includes compiling, linking, and packaging the code.

Selenium gives a single interface that allows one to write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others to test them.

Jenkins is a tool that helps the user continuously integrate and continuously deliver projects on any platform. It is an open-source application of DevOps.

Containerization is a type of virtualization in which all the components of an application are bundled into a single container image and can be run in isolated user space on the same shared operating system. Containers are lightweight, portable, and highly conducive to automation.²



- **Docker** is a platform that packages an application and all its dependencies together in containers. Containerization ensures that the application works in any environment. Containers can have a multitude of applications installed and pushed to the QA team, after which the QA team runs the container to replicate the developer's environment.

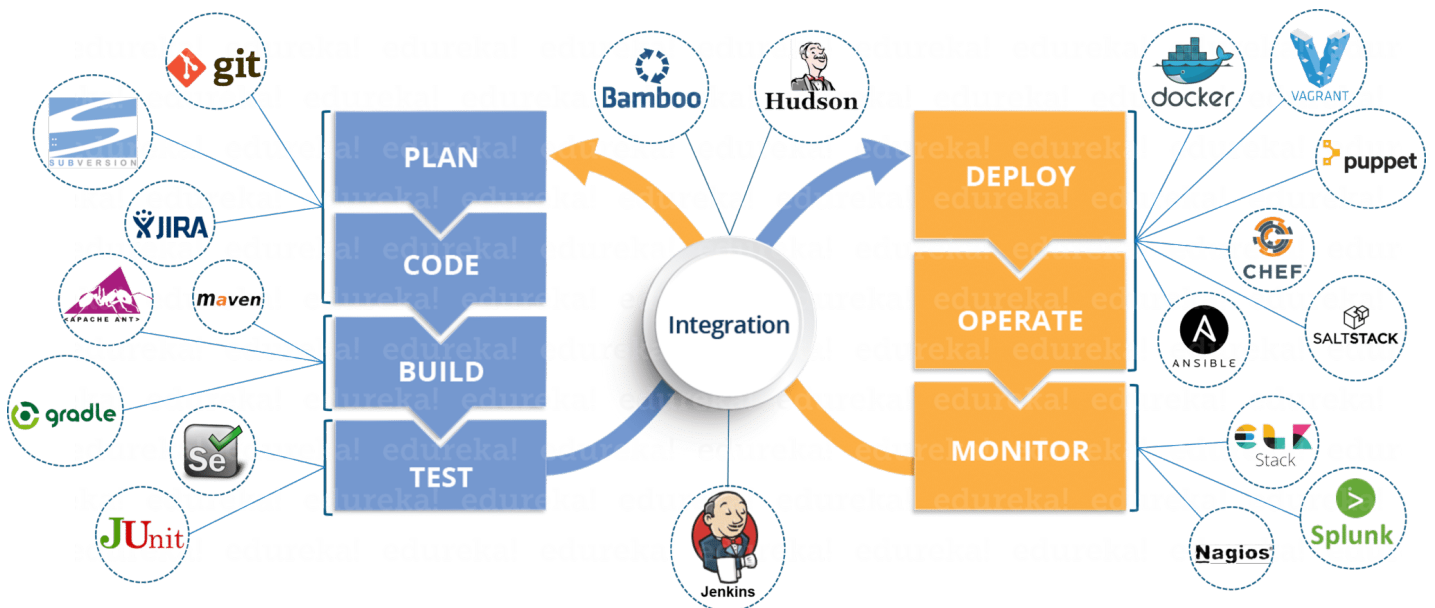
² Refer to no. 2 in references

- **Kubernetes** is an open-source system that handles scheduling containers onto a compute cluster and manages the workloads to ensure they run as the user intends. It works brilliantly with all the cloud providers to become a multi-container management solution.

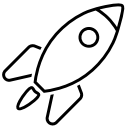



Maven is an open-source build tool developed to build, publish, and deploy several projects en masse for better project management. The tool helps developers to develop and document the entire lifecycle framework.³

BENEFITS OF DEVOPS

| | |
|--|---|
|  | <p>SPEED</p> <p>By leveraging DevOps, organizations can move their development lifecycle by innovating quickly, adjusting to shifting market needs, and increasing effectiveness, generating significant business results.</p> |
|  | <p>SWIFT DELIVERY</p> <p>Speeding up releases would ensure that products can be delivered to the end-users quicker to gain a competitive edge over other players.</p> |



³ Refer to no. 7 in references

| | |
|---|--|
|  | <p>RELIABILITY</p> <p>As continuous integration and continuous development are two pillars of DevOps, deployment of application updates and changes in infrastructure can be enforced reliably and efficiently while providing end-users with a consistent experience. Monitoring practices offer real-time insight into performance.</p> |
|  | <p>IMPROVED COLLABORATION</p> <p>The development and operations teams work together at every step, promoting shared responsibility and accountability and combining all workflows. This would ensure that inefficiencies that arise are reduced considerably while saving time.</p> |
|  | <p>SECURITY</p> <p>Bringing forth automation and integration of security testing solutions, such as organization-wide compliance policies and configuration management techniques, can help track compliance violations at any scale.</p> |
|  | <p>SCALABILITY</p> <p>Automation and consistency can help mitigate complex issues with reduced risk spanning across development, production, and testing phases repetitively and efficiently.</p> |

INDUSTRY PRACTICES USING AZURE DEVOPS

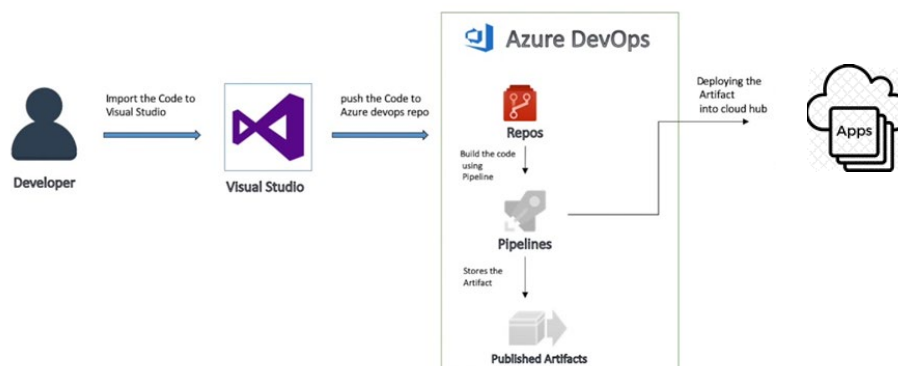
Cloud computing is gradually becoming the primary staple source for every business. It provides the required security and safeguards unnecessary expenses, overheads, etc. Designing and developing cloud applications is an excellent advantage to any organization.

One such entrant is Microsoft, with its Azure platform. Microsoft Azure is a cloud platform that assists in this endeavor by constantly expanding its potential. Developers can scale according to the user's requirements. Azure is compatible with all major operating systems, programming languages, frameworks, and tools, making it an enticing platform to work with.

Some of the tools offered by Microsoft Azure that overlay the DevOps environment are as follows. ⁴

- **Azure Ping:** This is a development tool designed to bring effective productivity. This is a utility extension that runs as a local Windows Service. Azure Ping helps in keeping tabs on Azure Storage resources.
- **Cloud Explorer for VS:** This tool helps in the identification of Microsoft Azure resources directly from Visual Studio IDE (Integrated Development Environment).
- **Azure Blob Studio:** This tool helps in managing Azure blob storage by creating containers, retrieving URLs and managing blobs, which is beneficial to developers.
- **Azure Grid:** The Azure Grid framework helps with more accessible code definition, grid establishment and allocation of data for each node. This translates to better control through a WPF client application.
- **Docker:** Docker is one of the crucial tools for Microsoft Azure developers. It is presently the recommended option for creating, managing, and deploying application containers. Docker is a wholly supported tool with Microsoft Azure.

Azure DevOps workflow



- **Microsoft Azure Storage Connected Services:** Microsoft Azure Storage Connected Service is a new tool from Microsoft that helps access design-time tools. The design-time tools can help configure Azure storage services and generate code for the consumption of services.

⁴ DevOps workflow model using Azure

DEVSECOPS

Another feature within DevOps is to prioritize security while developing software. Despite efforts by companies, security breaches still occur. Since 2000, over 3.5 billion people have seen their personal data being compromised. Part of the problem is that as application code scales and its complexity increases, so does the surface area for security vulnerabilities.

Organizations have been adopting a DevOps approach to automate the integration and deployment of processes across different teams, because of which traditional security tools often no longer suffice.

Developers need to introduce security into each step of the DevOps workflow; hence the process is often referred to as DevSecOps.

5

WHAT IS DEVSECOPS?

DevSecOps integrates security into continuous integration, continuous delivery, and continuous deployment. By involving DevOps values in software security, security verification becomes an integral part of the development process.

Like DevOps, DevSecOps integrates IT tools and automates project management workflows. DevSecOps integrates operational security audits and testing into DevOps workflows so that the product has security built into each step rather than applied to a finished product.



⁵ DevSecOps workflow

Implementing DevSecOps involves-

- Introducing security throughout the software development lifecycle to reduce the surface area of vulnerabilities.
- Security best practices are included in the DevOps team, including Developers and Operation Teams.
- Automated security checks are performed at each stage of the software delivery through various means, which span tools, processes, and security controls in the DevOps workflow.
- Every stage with automated security checks by integrating security controls and tools into the DevOps lifecycle.

PRESENT-DAY UTILIZATION PROS AND CONS

Like every other field in the world of IT, DevOps has its share of pros and cons. Some of the known advantages and disadvantages of DevOps have been explored below.

ADVANTAGES OF DEVOPS

- **Faster product release and time to market:** DevOps is often sold as cloud-based services associated with easy-use tools. This provides separate operations; hence a developer would have to focus on only one tool without understanding the entire toolchain. This improves better turnaround and collaboration.
- **Higher productivity and process efficiency:** The process is simplified with the indulgence of limited focus areas for the developers and operations. Hence the existing process has better productivity and efficiency, thereby even reducing the production cycle.
- **Engaged and motivated staff results in better Customer experience management** as the team has a fixed and focused task that is set in hand. This engagement also gives the team a clear product vision.
- Finally, with a practical and motivated workforce working with high efficiency, we can expect increased deployment success rates, product quality, and flexibility in support releases. This also reduces the chances of product failures.
- The cyclic approach works as a constant feedback system that helps the team always remain focused on self-improvement, making it a win-win scenario.

DRAWBACKS OF DEVOPS

- **Cloud Complexity and Compatibility:** Since the production environment is in the cloud, the infrastructure test process is complex. One would require an in-depth understanding of integration and orchestration workflows. The DevOps methodology focuses mainly on speed over security.
- **Demands proper mindset across the company on knowing what is required and the limitations.** Since speed is prioritized over security, one tends to lower business security by outsourcing DevOps operations. This is altogether another issue when dealing with continuous integration and continuous delivery. Organizations must aim at being fault-tolerant and alleviate human error. Automation of testing processes could be enforced and require a highly skilled engineering team to accomplish the goal.
- **Dealing with the legacy systems in the production environment, i.e., getting the right pool of expertise to implement DevOps, is challenging.** In addition to the costs for new hardware and software, organizations must recruit specifically skilled engineers, developers, and operations personnel, increasing the complexity of implementing the model and potentially risking downtime.
- **Organizations must be ready to face the organizational and technical changes required during the transition.** This includes transitioning to new DevOps tools and moving away from today's tools.

FUTURE SCOPE OF DEVOPS

The future scope of DevOps is heavily reliant on maximizing the outcomes while mitigating and staying vigilant of the disadvantages, which can be considered insights for improvement.

DevOps, undoubtedly, has brought about a palpable culture shift worldwide. The world has undergone a digital transformation, and companies are pressured to innovate faster for their teams and customers and to stand apart from their competitors. Some of the applications of DevOps in various industry segments have been explored:

IT Industry: Organizations like Amazon (through AWS), HP, Netflix, Adobe, and Atlassian, among others, have already adopted DevOps models in their operations. The IT Industry streamlines operations and enables developers to release features much faster without impacting their customers.

Banking: The banking and finance sector often relies on DevOps for their IT infrastructure as it complies with regulations like the PCI DSS (Payment Card Industry Data Security Standard) and manages the confidential information of its end-users. In addition, they can use DevOps by getting streamlined feedback loops and frequent software releases of their banking applications, enhancing the user experience.

Healthcare: As healthcare catalogs sensitive patient information and complies with several federal laws, DevOps toolchains can manage data through efficient data pipelines and large data applications. This is relevant to medical device testing reports, lab reports for patients, electronic health records, wearables, insurance claims, etc., and leverages them for audit purposes.

Inventory management: DevOps is an enterprise-wide set of rules that can be enforced in a wide range of businesses, including sales and marketing, customer service, quality assurance, IT operations management, project management, quality assurance, and human resources to streamline all activities in a much-reduced time frame.

CONCLUSION

In a world where developers are desperate to keep up with the need for faster, more frequent updates in an ever-changing industry, constant updates are not customer-facing alone. Businesses demand faster, better products to help them be more competitive in their respective industries.

DevOps is one of the new approaches to software development that is gaining precedence in the industry. DevOps has significantly impacted building a concrete structure behind designing any cloud-based application and is widely used across all sectors. The recent advancements in this field have spread over a large area like a forest fire leading to vast research openings unseen before.

The above article has covered all points and aspects one should know before choosing DevOps. The unwritten truth is that DevOps has a bright future. With more companies readily accepting this methodology as its standard and with constant evolution of new tools and technologies, DevOps is undoubtedly the go-to technology path.

The authors of this article sincerely hope that the article and the topics discussed will answer all your questions and give you an end-to-end overview of DevOps.

REFERENCES

1. C. Ebert, G. Gallardo, J. Hernantes and N. Serrano, "DevOps," in IEEE Software, vol. 33, no. 3, pp. 94-100, May-June 2016, DOI: 10.1109/MS.2016.68.
2. Containerization- "What Is Container Security? - Check Point Software." *Check Point Software*, Accessed 17 Jan. 2023. <https://www.checkpoint.com/cyber-hub/cloud-security/what-is-container-security/what-is-containerization/>
3. Azure DevOps- "What Is Azure DevOps and Why Should We Use It? - Super sourcing." *Super sourcing*, 13 Feb. 2022, super sourcing. com log/what-is-azure-devops-and-why-should-we-use-it.
4. "A Brief History of Devops [With Infographic]." *A Brief History of Devops [With Infographic]*, www.knowledgehut.com/blog/devops/history-of-devops. Accessed 17 Jan. 2023.
5. "Why You Need DevOps — and How to Get It Right - Near Form." *Near Form Enterprise Software Solution Development*, 12 Jan. 2021, www.nearform.com/blog/why-you-need-devops-and-how-to-get-it-right.
6. "Containerization - Wikipedia." *Containerization - Wikipedia*, 1 Sept. 2017, en.wikipedia.org/wiki/Containerization.
7. Katz, Eyal. "The Essential Steps to Building Your Own DevOps Pipeline." *The Essential Steps to Building Your Own DevOps Pipeline*, 22 Mar. 2001, blog.exigence.io/build-devops-pipeline. <https://blog.exigence.io/build-devops-pipeline>

Disclaimer: The views, processes or methodologies published in this article are those of the authors. They do not necessarily reflect Dell Technologies' views, processes, or methodologies.

Dell Technologies believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." DELL TECHNOLOGIES MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying and distribution of any Dell Technologies software described in this publication requires an applicable software license.

© 2023 Dell Inc. or its subsidiaries. All Rights Reserved. Dell and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.