



Your Comprehensive DevOps Guide

White Paper



Introduction

DevOps helps in transformation. It helps in organizational improvement. DevOps ensures reliability. Need flexibility? Go with DevOps.

It is most likely that you have heard one or all of these arguments in favor of DevOps. But what exactly is DevOps? Is it a set of tools? Does it bring in a fundamental change in the way teams work? Is DevOps a culture? Does it mean a completely new team?

While the DevOps adoption and practices have matured and DevOps is set to become a standard way of working, there is still some confusion in the minds of enterprises as to what exactly is DevOps and how to make it a success.

In this comprehensive guide by InfoBeans, we try to answer many of these questions. In this guide, we will know –

- What is DevOps and what business benefits does it offer?
- The difference between Agile and DevOps
- The key pillars of DevOps
- Best practices to follow to ensure your DevOps success
- An introduction to some of the popular DevOps tools

What is DevOps

In simple terms, DevOps is the practice where the operations and engineering teams together participate in the end-to-end product lifecycle – right from design, development to production support.

DevOps removes the silos of development and operations teams and merges these two teams into a single team where each team member work across the application lifecycle and possesses a wide range of skills.



DevOps essentially is an ethos - Through a combination of cultural change, processes, practices, philosophies, tools, and technologies, it helps enterprises in the delivery of applications at a much higher velocity (as compared to traditional software development processes), enables them to better serve their customers and achieve faster time to market.

Business Benefits of DevOps

The high-performing IT organizations adopting DevOps practices are deploying 200 times more frequently and with 2,555 times faster lead times than their low-performing counterparts.¹

DevOps helps organizations become more productive and profitable. Let us look at some of the key business benefits of using DevOps -

- **Better Quality and Reliability**

In the user's minds, the quality of a product or a service is measured in terms of the availability of the service and how fast the service is brought back in case of disruption. DevOps allows organizations to excel on both these fronts. Because of faster feedback loops and ability to make frequent releases, the deficiencies can be removed quickly allowing the organizations to improve the overall service quality and reliability. Since frequent releases can be done with short lead times, it ensures user satisfaction through timely delivery.

- **Improved Collaboration**

DevOps encourages tight collaboration amongst the team members. It places a lot of importance on accountability and ownership. It encourages the development and operations teams to share responsibilities and work closely to reduce inefficiencies and save time. It promotes the culture of constant learning and encourages team members to rely on each other. Such fresh perspectives and cultural changes help in improving the output and productivity of the team members.



- **Faster Delivery and Increased Responsiveness**

DevOps enables the organizations to quickly adapt to the changing market conditions, innovate faster, and drive better business results through faster time to market. It speeds up the delivery cycles by removing the unnecessary processes and handovers between the teams. Through shorter release cycles and continuous customer feedback, DevOps helps organizations in delivering true value of the service to the users.

- **Better Security**

With automated compliance policies, better controls and configurations, DevOps allows organizations to achieve faster time to market without compromising on the security aspect.

- **Increased Employee Satisfaction**

DevOps emphasizes on close collaboration within the team. Since all the team members are involved in the complete delivery cycle, they get a holistic view of the complete service delivery cycle. It gives each team member a sense of accomplishment as everyone gets to know where and how exactly their piece fits into the whole system. The personal collaboration amongst the team members helps in building a healthy team spirit.

DevOps Vs Agile – What's the Real Difference

Agile and DevOps are the buzzwords in the industry today but these are not the same. Let us quickly understand the differences between these two software development methodologies

- DevOps is about software development and management, whereas; agile is only about software development.
- Agile teams, in many cases, follow SCRUM implementation method. In this, there are daily standup meetings to update the team members on progress and bottlenecks. Agile team does not always have defined designed documents and specs. On the other hand, DevOps team members don't necessarily interact on a daily basis and tend to rely more on specs and design documentation.



- Agile teams are generally smaller. DevOps team, by definition, has multiple teams working together as one team.
- Documentation does not have a very crucial role to play in Agile and the teams are more flexible with the features to be included in each release. DevOps relies heavily on documentation.
- DevOps relies very heavily on automation whereas; Agile teams can choose to use certain specific automation tools or completely stay away from automation.

Key Pillars of DevOps

The key to DevOps success is tight collaboration and coordination between the business, development, QA, and operations teams throughout the application lifecycle. Let us take a look at the key pillars of DevOps.

Automation

While DevOps is not only about automation, DevOps relies heavily on automation for development, testing, and deployment – so much so that a certain tool form an integral part of DevOps. The key lies in striking the fine balance with automation. Different stakeholders can use different tools which can be brought together through automation and an end-to-end automated DevOps process can be created. Automation helps in better collaboration between the teams and helps in overall service lifecycle management.

Continuous Integration (CI)

Continuous Integration is a software development practice which encourages immediate testing of isolated code development when it is added to a larger code base. The objective of CI is to quickly detect and fix bugs and improve software quality while ensuring timely delivery. Since the code gets integrated on a frequent basis, the integration issues and conflicts can be identified and fixed before they become too significant. This also fosters frequent and on-going communication and collaboration between the developers and improves collaboration.



Continuous Testing

Continuous Testing is one of the most crucial pieces of DevOps. In DevOps, Continuous Testing is the responsibility of everyone involved in the team. The delivery teams roll out small but constant updates which the QA teams test. The operations team also has a crucial role to play in testing – based on the load patterns and usage patterns in the production environment, the operations team helps in creating the right environment with the right configuration for load testing. The operations team actively participates in load, stress and functional testing by offering their guidance based on their experience of working with other similar applications.

Continuous Delivery

Continuous Delivery is essential for DevOps teams which are involved in iterative software delivery. In this approach, teams make sure that every change to the software can be released instantly. When implemented properly, Continuous Delivery ensures that developers always have a code which has passed through the test process and is ready for deployment. Automation plays a crucial role in Continuous Delivery - The code changes are automatically built, tested, and released to production. The aim of Continuous Delivery is to make frequent releases.

Continuous Monitoring

Owing to frequent releases, some bugs and defects can seep through the release. In a DevOps environment, the goal is to quickly find and fix the failures in real time through Continuous Monitoring. Continuous Monitoring allows organizations to quickly identify when there is an issue, find out the probable causes, and use the learning to avoid future issues. Typically, there are two types of monitoring: Server monitoring and performance monitoring. Effective monitoring requires a right set of tools which are deployed in development as well as the production environments.



Best Practices to Follow to Achieve DevOps Success

As we saw, DevOps is not just a set of tools or processes which can be deployed to achieve the status of “DevOps Compliant”. DevOps, in fact, is a complete shift in the organization culture and is based on People, Processes, and Tools. Here are some of the best practices to follow to ensure DevOps success within your organization -

- For every feature or new application under development, define the expected impact on areas such as security, architecture, network configuration etc. right in the beginning.
- Automate infrastructure provisioning to enhance development and testing speed
- Mimic production environment as much as you can – right from the apps, data, user devices etc. to effectively test all integrations.
- Enforce continuous integration to identify issues which need to be acted upon early in the development cycle.
- Implement test automation to test various application features, verify code quality, and test the stability of the overall solution.
- Implement thorough tracking mechanism which tracks not only the defects but also the changes in code, features, environment as well as the business requirements.
- Orchestrate deployment process to ensure reliability and consistency in application behavior and enhance the deployment efficiency.
- Shorten the delivery cycle using process automation and scope minimization to increase the speed of delivery of new features.
- Leverage monitoring to identify the deviation patterns and use this data to proactively address issues and reduce response times.



An Introduction to Popular DevOps Tools

Here are some of our favorite tools used in DevOps-

✓ CI Tool -

Jenkins is one of the most popular tools for doing continuous integration on any platform or technology. It is widely used for automation of software building, testing, and deployment. It allows integration with several testing and deployment technologies.

✓ CD Tool -

Deployer is a deployment tool written in PHP language. It allows the creation of deployment scripts based on readymade modular blocks called "recipes".

✓ Code Analysis Tools -

PHPLint: It is used to verify PHP syntax check.

PHPMD: It is used to detect mess in the code such as unused variables, unused functions or unused parameters, sub optional code, over-compiled code etc.

PHPCS: It is used to find out violations against defined coding standards.

PHPUNIT: This framework, written in PHP language, is used to unit test PHP code.

PHPCPD: This copy paste detector detects the redundant or same code.

✓ Docker -

Docker is an open source project for developing, shipping, and running applications. Docker provides containers where you can run an application in the isolated environment. On a given host, one can run as many containers as required. Since it is lightweight, it uses less RAM and starts instantly.

✓ Vagrant -

Vagrant automates virtual machine creation. It is used for creating and managing lightweight, reproducible, and portable virtual machine environments.



✓ **Ansible -**

This general-purpose automation tool is used for configuration management or workflow automation. Ansible allows configuration of multiple machines in less time.

✓ **Git -**

It is used as a distributed version control system.

✓ **Robot Framework -**

This generic test automation framework is used for functional testing.

Conclusion

“DevOps”, the term first coined in 2009 by Patrick Debois, has almost become a standard practice in software development today. DevOps helps enterprises in faster and more frequent software deployments without hampering the stability and quality of the system. It, thus, helps them in being more competitive and innovate faster. DevOps promotes shared responsibilities and trust within the development and operations teams. It extends the continuous development goals to continuous integration and continuous delivery. To facilitate frequent releases, DevOps leverages automation extensively. To ensure success with DevOps, organizations need to bring in a fundamental cultural change in process, people, and technologies that they use for software development and deployment.

“DevOps is firmly in the mainstream today...DevOps is no longer a nice to have or a differentiator but is now essential to compete effectively in the market.”



About InfoBeans

InfoBeans is a technology service provider offering development and implementation of cutting-edge software solutions for various small and large enterprises across all verticals. With our extremely innovative, dedicated and experienced team, we have helped organizations worldwide in developing robust and ascendable solutions.

InfoBeans has worked with enterprises in the automobile, engineering, telecom, currency printing, and storage domains and helped them with their custom software development, storage and virtualization, UI/UX, eCommerce, and automation engineering needs.

Our team of experts specialize in build and release automation that enables businesses to achieve continuous integration and faster deployment cycles.

For more information, visit us at <http://www.infobeans.com>

References

1. The 2016 State of the DevOps Report
2. Ovum Research; Changes Within and Driven by DevOps; June 2016

----- End of Document -----