

# DEVOPS

**A Successful Path  
To  
Continuous Integration  
And  
Continuous Delivery**



# INDEX

<b>01. Introduction</b>	<b>02</b>
<b>02. DevOps By Definition</b>	<b>04</b>
<b>03. A Complementary to Agile Process</b>	<b>05</b>
<b>04. The DevOps Lifecycle</b>	<b>06</b>
<b>05. DevOps Patterns</b>	<b>07</b>
<b>06. Key DevOps Components – A Glance at CI/CD Process</b>	<b>08</b>
<b>07. DevOps and Automation</b>	<b>10</b>
<b>08. DevOps Value-addition to Enterprises</b>	<b>11</b>
<b>09. Key DevOps Benefits – A Snapshot</b>	<b>12</b>
<b>10. DevOps Open Source Tools</b>	<b>13</b>
» 10.1 DevOps Tools For Continuous Integration, Continuous Management	14
» 10.1.1. Chef	14
» 10.1.2 Puppet	14
» 10.1.3 Ubuntu Juju	14
» 10.1.4 Ansible	14
» 10.1.5 Rancid	14
» 10.2 DevOps Tools For Continuous Delivery	15
» 10.2.1 Fabric	15
» 10.2.2 Capistrano	15
» 10.2.3 Jenkins	15
» 10.3 DevOps Tools For Continuous Testing	16
» 10.3.1 Ant	16
» 10.3.2 Gradle	16
» 10.3.3 Maven	16
» 10.3.4 BuildHive	16
» 10.4 DevOps Tools For Continuous Monitoring	17
» 10.4.1 New Relic APM	17
» 10.4.2 New Relic Server	17
<b>11. Conclusion</b>	<b>18</b>

# INTRODUCTION

In today's world, business needs to be effectively agile to meet the ever-changing market needs and balance diverse consumer demands. While business innovation is largely driven by software in this digital age, it is important for enterprises to ensure faster release of application updates (without compromise on quality) to win, serve and retain customers.

Traditionally, enterprises focused on delivering error-free products to customers. Now, the objective has changed to providing more value to the business with faster release cycles and ensuring quicker RoI.

This White Paper details an insight into various underlying concepts and facets of **DevOps** that strive for Continuous Integration (CI) and Continuous Delivery (CD), and also answers the key question: **'Why enterprises need DevOps?'**



The introduction of **cloud computing** led to the addition of three unique traits to the enterprise IT mix, in the form of new application platform, new model for consuming IT services and a 'strong foundation for new business models.

Thus, a truly innovative idea, robust technology and easy implementation have become need of the hour for enterprises, giving rise to the combination of Development (Dev) and Operations (Ops) teams as 'DevOps'!

DevOps is gaining popularity for its ability to focus on managing, tracking and automated software releases across the entire application lifecycle. With this, Dev and Ops teams began redefining their approach towards collaboration to align their goals, priorities and be able to adopt shared toolsets, which DevOps innovation made possible!



# DevOps By Definition

DevOps can be defined as a culture that primarily focuses on improved collaboration, communication and integration between Dev and Ops teams. It is an umbrella term that is also described as a philosophy and cultural change that paves way for a paradigm shift.

Dev and Ops are the two major entities involved in successful delivery of a software or to make a deployment of high quality. DevOps promptly breaks the existing barriers between the two and brings about a cohesive, seamless functioning across organizations.

DevOps principles not only enhance the performance of software development and operations, but also show a positive impact on the web service development and quality assurance aspects. DevOps lays a path for a culture of knowledge and information sharing that bridges communication gaps among IT teams, thus facilitating establishment and continuous delivery.

Many globally-renowned research firms indicate a continuously-growing path for DevOps since its inception and towards the future!

“By 2020, DevOps will evolve from a niche to a mainstream strategy employed by 25 percent of Global 2000 organizations,” Gartner said in a report.

# A Complementary to Agile Process

In the last few years, the agile software development moved towards a new practice under the label 'DevOps'.

While agile software development focuses on the collaboration between the business and its developers, DevOps focuses on collaboration between developers, IT operations and security teams. Agile software development provides business agility, while DevOps provides IT agility, enabling the deployment of reliable and predictable applications with shorter release cycles.

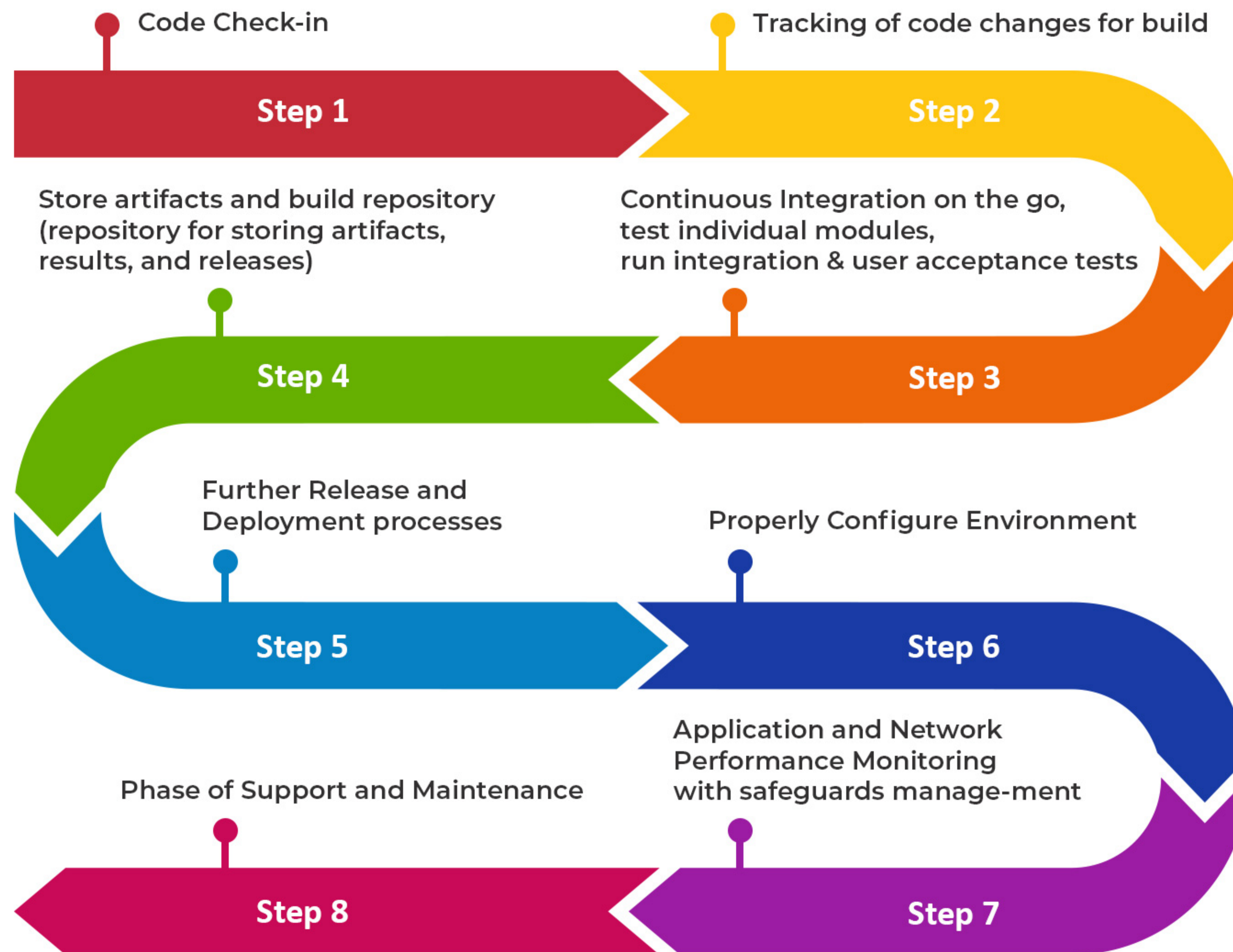
DevOps can be termed as 'complementary' to agile software development, as it extends continuous integration and continuous release by making the code 'production-ready with high value to customer'.

A disciplined agile delivery is an established process for developing software and it includes DevOps. Moreover, DevOps structurally breaks the difference and unifies Dev and Ops teams.

The goal of DevOps is to successfully deploy features into production without causing any disruption to other services, while quickly detecting and correcting incidents as and when they occur during the DevOps lifecycle.

In its first-ever DevOps and Application Performance Survey, IDC reports about downtime issues causing billions for large companies, around USD 1.25-2.5 billion every year for Fortune 1000 firms. Citing this, the report presents its first-ever real insights into the adoption and impact of DevOps practices in large enterprises as a solution!

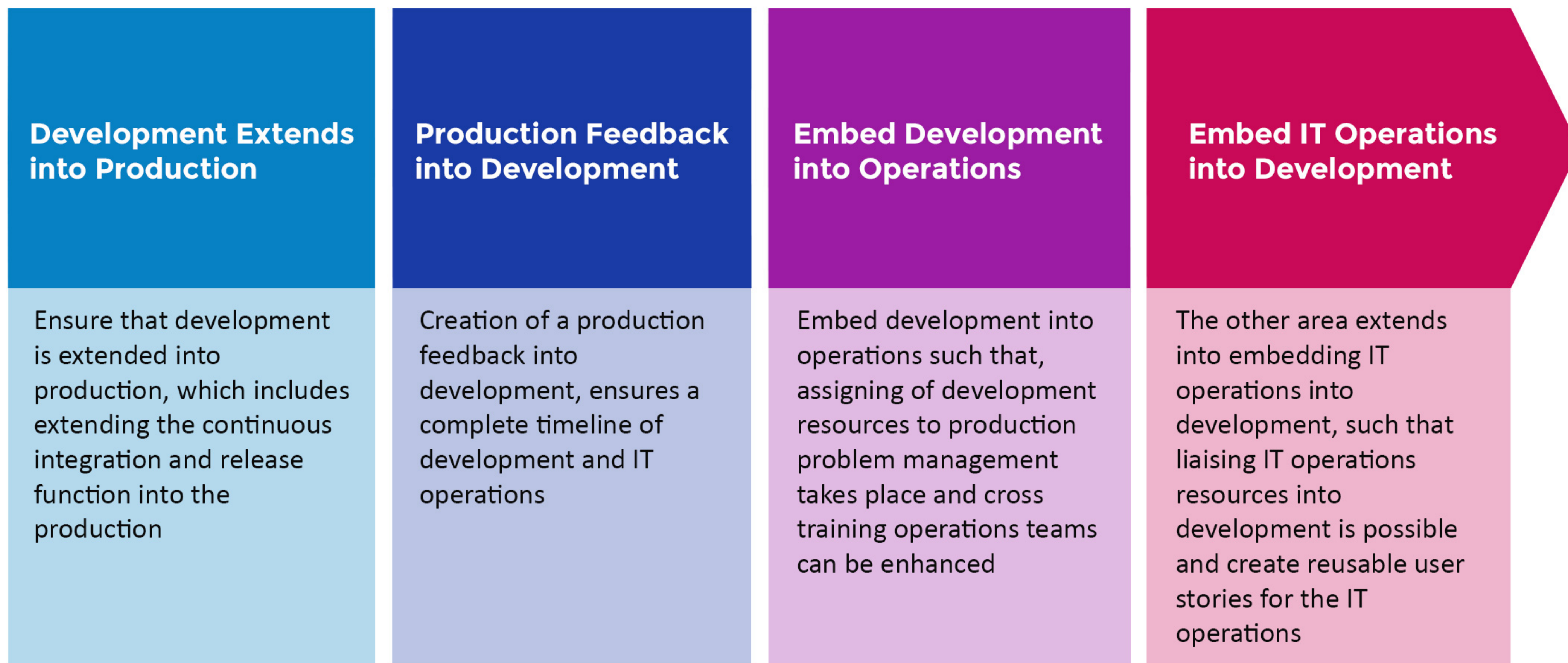
# The DevOps Lifecycle



DevOps Lifecycle enables continuous product deployment through effective infrastructure automation, configuration management, deployment automation and infrastructure monitoring, while also deriving an effective log management.

# DevOps Patterns

DevOps patterns can be categorized into four main areas, as depicted below:





# Key DevOps Components

## A Glance at CI/CD Process

Continuous Integration and Continuous Deployment form two key core aspects of DevOps.

**Continuous Integration (CI)** is a key benefit of agile development and DevOps processes.

Here, private builds developed by different developers (under a regular development environment) and tasks of other teams are delivered commonly to a team build server.

All the work is later integrated in a common build area to form an integration build. The process then moves across common cross-team build server and then applied system-wide or application-wide. This is how the continuous integrations and builds happen, making the CI pipeline possible.



Continuous Integration (CI) can be defined as a software engineering practice, where isolated changes are immediately tested and reported when they are added to a larger code base. The goal of CI is to ensure timely detection, addressal and feedback in case of any defect reported during the product lifecycle.

Thus, DevOps boosts deployment frequency and presents more opportunities to re-evaluate the delivery process, through automation, effective testing and monitoring procedures. DevOps practices provide valuable data for continuous improvement around monitoring and metrics.

However, CI should be a part of every DevOps process, irrespective of organizational size or scale, and should undoubtedly be driven by a strong Quality Assurance (QA/QC) strategy.

Coming to **Continuous Delivery (CD)**, DevOps relation with the CD pipeline revolves around the new features that developers work with and those released to customers, in a timely manner. All the builds that pass through QA need not go into production. Only those with functional stability will move to production and further to become 'production-ready' before staging.

The practice of regular delivery of applications (under development) to QA and Operations for validation, and potential release to customers is termed as a Continuous Delivery (CD).

Now, we will look into the key aspect of DevOps that has completely changed the way a process chain operates, i.e. **Automation**.

# DevOps and Automation



Automation forms the core practice of DevOps, wherein a special focus is laid on setup, configuration, deployment and assisting infrastructure and applications. Automation helps set up environments more rapidly in a standardized and automated manner.

- Earlier, server configuration and application deployment were predominantly manual processes with high vulnerability to errors, unreliability and inability to support agile business. To address these concerns, organizations were employing highly-skilled resources to provide manual configuration. But this did not solve the problem leaving scope for impact on critical, high-value activities such as software release, machine configuration, operating system patching, troubleshooting or bug fixing, within a business.

This is where Automation comes as a savior, automating majority of the critical business tasks!

DevOps environment presents a higher level end-to-end automated process, eliminating enterprise burden caused by manual intervention or access to the production environments. DevOps relies on tools to automate large parts of the end-to-end software development and deployment process.

# DevOps Value-Addition To Enterprises

DevOps is today a buzzword driving enterprises to realize quicker time-to-market. It empowers the teams with a cultural shift and enables an improved continuous flow in IT operations.

Traditional SDLC process was more of chunks of code delivered in frequent intervals, deployed at a later stage and deployments that often result in chaos and disruption.

DevOps practices came in collaborating developers, operations and other teams. In this process, developers devote some time to operations, helping the Ops team embrace some aspects of automation. Working with operations teams also helps developers understand how their applications run in production and also encourages them to improve logging, monitoring, deployment practices for faster or quicker releases.

A sort of best practice application delivery can be achieved with DevOps application delivery mechanism because of improved communication, collaboration and integration between development and operations teams.

Thus, DevOps is a pure Developer-Operations collaboration that provides improved process delivery, enhanced business agility and efficient integration for enterprise of any size.

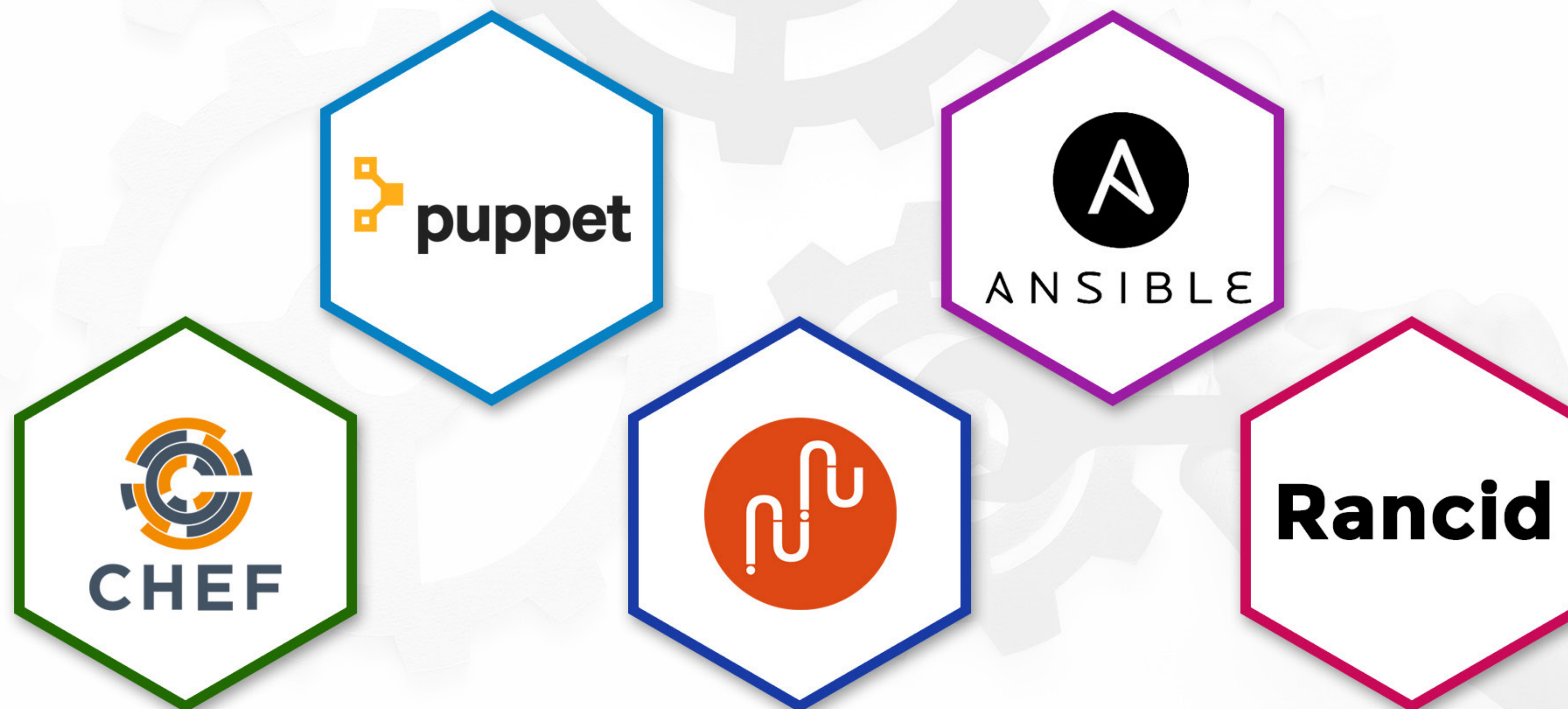
### **Key DevOps Benefits - A Snapshot**

1. Ensures stable and reliable operating environments
2. Facilitates continuous release and deployment
3. Improves quality and time for innovation
4. Treats infrastructure as code
5. Ensures smaller and faster deployments, ensuring faster time-to-market
6. Improves ROI, the key to any business success
7. Mitigates risk by reducing time to delivery
8. Identifies problems and provides smooth and effective resolution

# DevOps Open Source Tools

There are many commercial and open source testing tools that can help you check the performance of your applications. Open source tools have seen wider adoption in the DevOps era. Open source tools provide critical visibility and help resolve bottlenecks quickly.

It is also important to choose a right tool that fits in your business strategy, which obviously needs some thought and time. Below are important DevOps tools, both script-based or container-based models, that can you help in your DevOps implementation.



## DevOps Tools for Continuous Integration, Continuous Management

Some of the tools for Continuous Integration and Continuous Management include Chef, Puppet, Ubuntu Juju, Ansible, RANCID, detailed below:

**C** **hef** automates the cloud and data centers, enforces compliance and security, and ensures continuous delivery of applications and infrastructure. The tool enables cross-team collaboration and facilitates a quick adoption of continuous delivery.

**A** **nsible** is an open source DevOps software platform that provides configuration management and automation services. It also includes ad-hoc task execution and multi-node deployment. The modules work over both JSON and standard output and can be written in any language.

**R** **ancid** is a network management application released under a BSD style license. It is used to monitor a router's or generally a device's configuration including software and hardware cards, serial numbers and more, and uses CVS or subversion to maintain a history of changes.

**P** **uppet** is an open source configuration management tool written in Ruby. It is the leading platform for delivering and operating constantly modern software, no matter where it runs. Whether you need to enforce security policies, prove compliance, move to the cloud or adopt complete DevOps practices for continuous delivery, Puppet has a solution!

**U** **buntu Juju** is a powerful time-saving service orchestration tool. This tool helps you model, configure and manage services with Juju, and deploy code to all major public and private clouds with a few commands. Hundreds of pre-configured services are available in the Juju store.

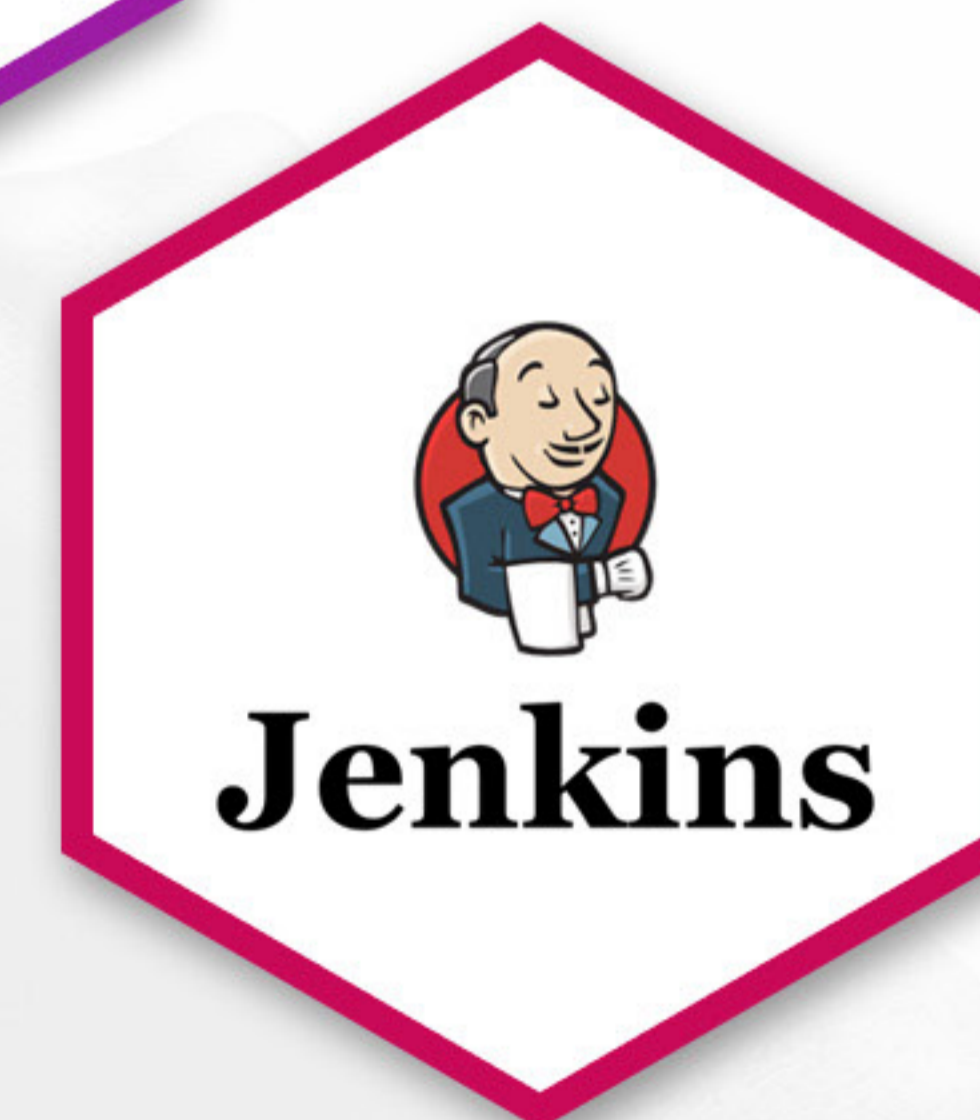
## DevOps Tools for Continuous Delivery

Fabric, Capistrano and Jenkins are some of the popular tools for Continuous Delivery, discussed below:

**F**abric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks. It provides a basic suite of operations for executing local or remote shell commands and uploading/downloading files as well as performing auxiliary functionality such as prompting the running user for input or aborting execution.

**J**enkins is a continuous integration and continuous delivery platform that makes it easy to continuously build and test software projects. It improves productivity by allowing developers to easily integrate changes and also helps users to access fresh builds without any hassles. Jenkins also integrates with a large number of testing and deployment technologies and provides ways to define your build pipelines to ensure continuous delivery of software projects.

**C**apistrano is a remote multi-server automation and deployment tool written in Ruby. This tool extends the Rake DSL with methods specific to running commands on servers.





## DevOps Tools for Continuous Testing

Some of the tools for Continuous Testing include Ant, Gradle, Maven, BuildHive, detailed below:

**A****NT** (Another Neat Tool) is a build tool created by Apache. This tool helps automate frequently-run tasks and presents an easy-to-use methodology that needs only an XML file with the definition of generic tasks for the job to be done.

**M****aven** is a tool used for automating build lifecycles using Java, C#, Ruby, Scala and other languages. Automation needs execution of only a small set of commands for building a project. Maven uses an XML file to explain the software project that is being built, along with its dependencies on plug-ins, external modules and components, etc.

**G****radle** is a tool that integrates the DevOps lifecycle. From the development in the Integrated Development Environment (IDE) through Continuous Integration (CI) server and from Docker containerization to deployment, Gradle integrates with everything you can imagine.

**B****uildHive** is Jenkins for the community and works with projects hosted on GitHub. Administrators log in to GitHub and can enable their projects for BuildHive with a click. It sniffs multiple project types - Ant, Maven, Gradle, SBT (Scala) and Rake (Ruby), and automatically sets up a corresponding build. In majority of the projects, users will end up with absolutely no configuration changes for their projects.



## DevOps Tools for Continuous Monitoring

Some of the tools for Continuous Monitoring include New Relic AMP and New Relic Server, detailed below:

**New Relic APM** helps teams to reduce resolution times, focus more on writing code and less on troubleshooting. End-to-end transaction tracing is possible across the entire service-oriented application environment with this tool. Drill down to see the performance impact of specific code segments and SQL statements.

Thus, enterprises moving towards DevOps need to rely heavily on automation, which requires tools. Though the potential benefits are compelling, there are many challenges for any enterprise or organization trying to successfully adopt a DevOps approach. Realizing those potential benefits can prove much tougher than expected and a failed implementation can be extremely disruptive.

This can be addressed with the right planning, mindset, technology and a good DevOps implementation partner.

**New Relic Server** helps you to take care of the entire monitoring checklist and ensures optimal server operations. It keeps the Dev, Ops and engineering teams on the same page and facilitates ways to deal with issues such as disk capacity and CPU, memory and disk I/O utilization.

The logo for New Relic APM is contained within a blue-outlined hexagon. It features the text "New Relic" in a black sans-serif font above the text "APM" in a larger, teal sans-serif font.

New Relic.  
APM

The New Relic logo is contained within a red-outlined hexagon. It consists of a teal circle with a white center, and the text "New Relic" in a black sans-serif font centered within the white circle.

New  
Relic.

# Conclusion

There are a plenty of tools available to help you in the DevOps journey. But, choosing a right one is a real challenge.

End-to-end automation requires the entire work to flow unimpeded across different teams. As organizations begin development with greater automation, they cannot jump immediately into delivery and deployment practices. If you are on the path towards DevOps, then this is an important aspect that must be executed correctly before any other aspect of DevOps can be implemented.

The market clearly perceives the benefits of DevOps tools and practices and presents many solutions that integrate with the leading categories of tools as well as legacy systems to provide end-to-end value.

It is not easy to implement DevOps in the enterprise, but with the right planning and careful consideration of your business needs, a business can surely witness the extremely-compelling benefits of DevOps with the right [DevOps implementation partner](#) in place!



info@veritis.com



1-877-VERITIS (283-7484)



972-753-0033



### US Corporate Headquarters

1231 Greenway Drive  
Suite 1040  
Irving, TX 75038



### India Headquarters

#607, 6th Floor,  
Ashoka Bhoopal Chambers,  
S P Road, Begumpet  
Hyderabad - 500003, India.  
Phone no: 040 42211730

For more information, contact [info@veritis.com](mailto:info@veritis.com)

© Copyright 2019 Veritis Group, Inc. All rights reserved. Veritis Group, the Veritis Group logo, and other Veritis Group marks are trademarks and /or registered trademarks of Veritis Group, Inc., in the United States and/or other countries. Other company and product names may be trademarks of the respective companies with which they are associated.