



# Your guide to getting started with DevOps

# Your guide to getting started with DevOps



## Contents

<b>Introduction</b>	3
Getting started on your DevOps journey	3
Part one	
<b>DevOps: an overview</b>	4
What is DevOps?	4
How DevOps can benefit your business	6
Part two	
<b>Fundamental technologies and processes</b>	8
The stages of DevOps processes	8
What powers DevOps stages?	10
Part three	
<b>Other key DevOps technologies you should understand</b>	13
Part four	
<b>Problems solved by DevOps</b>	16
Summary	
<b>Ready to get started?</b>	18



# Introduction

## Getting started on your DevOps journey

In a relatively short time period, DevOps has established itself as an essential software development methodology. In the quest for successful digital transformation, many are enticed by the benefits of a DevOps approach. Among the benefits, successful DevOps adoption drives faster, more dependable software deployment and delivery, increased operational efficiency and reduced security and compliance risks.

This flourishing IT philosophy can transform an organisation's ability to deliver applications and services at pace and scale. As well as a tech phenomenon, it's now a strong career prospect, accessible to a broad spectrum of IT professionals.

DevOps rapid rise to the top has meant many organisations and professionals occupying the IT space need to get to grips with, well, everything related to DevOps – and fast. Luckily, to help you get started on your DevOps journey, there are plenty of tools and guides available – like this one.

If you want to implement DevOps into your work structure and you'd like to delve deeper into the inner workings, technologies and processes, then you've come to the right place. In this eBook we'll start at the very beginning with the meaning of DevOps and discuss potential impacts and advantages to your business. We'll investigate DevOps technologies and processes and highlight some problems solved by DevOps. If you're ready to get started, let's jump in.



Part one:

# DevOps: an overview

## What is DevOps?

DevOps—the unification of development (Dev) and operations (Ops) teams—is a software development methodology that integrates and automates processes between development and IT teams through a combination of cultural philosophies, tools and practices. This approach is all about team empowerment, cross-team communication and collaboration and technology automation. A successful DevOps approach will enable organisations to develop and deliver applications and services at a faster and more efficient rate.

The DevOps movement can be traced back to around 2007. Software developers voiced a need for a more streamlined and efficient software development model. The traditional software model, where developers who wrote code worked apart from operations who deployed and supported the code, supported siloed processes, which resulted in bottlenecks and delays.

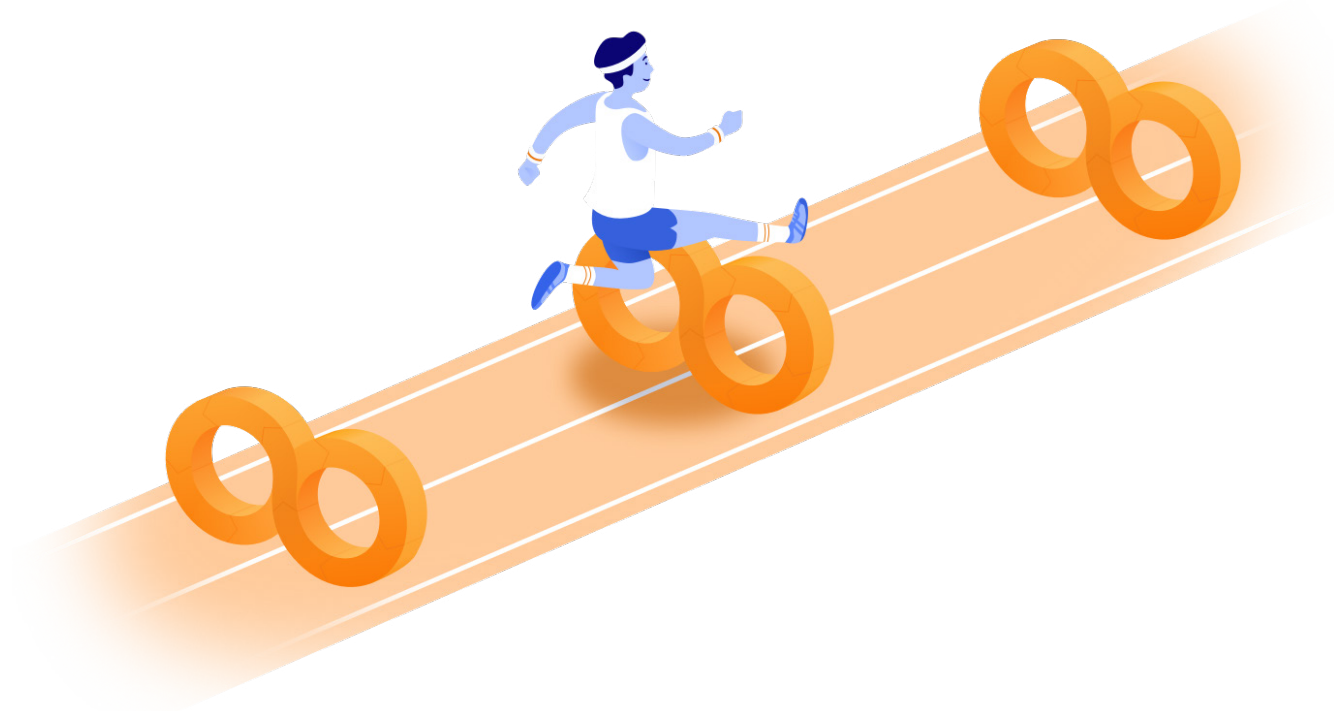
Under the traditional software development model, security was an afterthought rather than a vital step in the development process. Security breaches were dealt with after they had already taken place, as opposed to preventing them from happening in the first place. With the old software development model proving to be ineffective and the need for speed and a stronger focus on security, DevOps stepped into the spotlight.

And here's why.

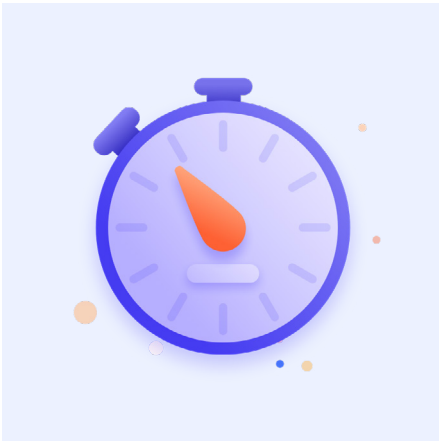
Behind any successful DevOps practice is a culture that promotes collaboration, communication and shared responsibility. People, processes and tools align, and cross-functional teams all take accountability for the complete product lifecycle. DevOps breaks down silo mentality. Instead of different teams working as separate entities, DevOps promotes a unified vision, where everyone works towards common goals.

What's more, with a single end-to-end DevOps platform, or one ecosystem, teams can streamline and simplify the number of complex tools used. As a result, teams and organisations become more efficient, as they save time otherwise spent jumping from tool to tool.

By using an ecosystem to conceive, build and deliver better, more compliant and secure software more efficiently, continuously and at top speed, DevOps teams and the wider business will become more agile. Businesses can more quickly meet customer needs, meet compliance standards and stay competitive in a changing business climate. But what are some of the other ways DevOps can help your business? Let's examine these in more detail.



## How DevOps can benefit your business



### **Speed**

Through automated tools, a “shift-left” approach to security and repeatable and measurable processes, a DevOps platform leads to better software delivered at pace. Teams can release deliverables more frequently while reducing the time between designing new, quality features and rolling them out into production.



### **Improved collaboration**

DevOps philosophy is all about building a culture of collaboration and communication between developers and operations teams. Teams work together rather than in silos, collaborate regularly and share responsibility for important factors such as security. Working together in a more collaborative way improves vulnerability detection and response, reduces potential downtime and enables your teams to better detect security issues.



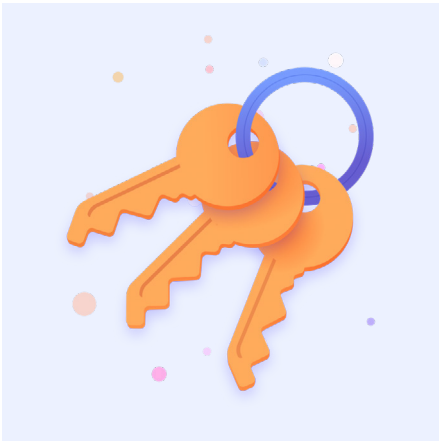
### **Rapid deployment**

DevOps helps organisations to increase the frequency and velocity of software delivery. By automating and streamlining the software development and delivery process, DevOps teams can improve products rapidly and ultimately deliver high-quality software faster and more consistently. This can lead to faster time-to-market, improved customer satisfaction and establishing a competitive advantage.



## **Quality and reliability**

DevOps practices such as continuous integration and continuous delivery ensures changes are functional and safe, which improves the quality of a software product. Plus, through monitoring, teams are provided with continuous real-time performance updates.



## **Security**

Security is an active, integrated part of the development process. It's fused into a continuous integration, continuous delivery and continuous deployment pipeline, so that it is a constant thread running throughout the process. Active security audits and security testing are integrated into agile development and DevOps workflows, so that security is built into the product.



Part two:

## Fundamental technologies and processes

### The stages of DevOps processes

The lifecycle stages of DevOps range from planning to performing analysis and gaining feedback. Understanding the flow of DevOps is a key part of creating efficiency, reliability, speed and agility. You can immediately access information created in any of the DevOps stages, using one platform as a single source of truth to manage and control the software development lifecycle. Working with a united platform enables better visibility and enhanced collaboration.

At a distance, there are three main stages: build, test and deploy to reflect the natural order - i.e. build the code, then test and deploy it. However, if we look more closely, we will find more layers are attached to these three stages. All in all, there are nine key stages that you need to know.

The stages operate in an ongoing cycle, with security woven into every stage of the process. When security is not an afterthought and runs throughout the process (as in a DevOps approach), issues, vulnerabilities and bugs are dealt with as they occur.





The stages are outlined below:

- **Plan** – this stage of DevOps takes place before any action, including before the first line of code is written. In this stage, product roadmaps are created to guide the upcoming development. It's also in this stage that organising resources, outlining priorities, aligning goals and setting up the tracking of projects takes place.
- **Create** – as the first stage of the CI/CD pipeline, code is designed and developed in this phase. This takes place using version control to coordinate changes made by multiple developers to the same code base. It's an important factor for improving speed.
- **Verify** – during this stage you will confirm the quality of code. Security testing, code quality analysis, parallel execution and automation all feature in this phase. This stage enables developers to find and fix flaws while they are developing, which is a cost-effective and more efficient method.
- **Package** – after code has been created and tested, this stage is where applications and dependencies are packaged and containers are managed to maintain a consistent software supply chain.
- **Release (also known as deployment)** – it's during this stage that updates are driven into the production environment. Code updates can be released as and when iterations are created, tested and ready (as opposed to a planned release date).
- **Configure** – this stage involves establishing and maintaining application environments across servers, networks and storage systems.
- **Monitor** – this proactive and automated part of the process strengthens agility, reliability and security. It focuses on software tracking and raising the alert to any problems.
- **Protect** – this stage is about bolstering the security of your applications and protecting their runtime environment from intrusions and exposures.
- **Manage** – the final stage establishes standard DevOps build and deployment practices. Here you will manage permissions and automate guardrails to adhere to compliance procedures and meet security policies. This stage is about creating visibility and control across your end-to-end software development lifecycle.

## What powers DevOps stages?

### Source Code Management (SCM)

SCM is used to track modifications to a source code repository. It plays a crucial role in easing the organisational burden of growing development costs. It tracks a running history of changes to a code base, which helps resolve conflicts when merging updates from multiple contributors.

Code is separated and managed into projects and groups of projects. An individual developer checks out existing code or adds code to what's there and the SCM tool identifies conflicting edits to the same code and flags it for resolution. It's a process which enables many developers to work on one project, ultimately increasing the speed of software updates.

### Continuous integration (CI)

This stage is a primary DevOps best practice and central to the entire lifecycle. CI is the practice of automating the integration of code changes from numerous contributors into a single software project. Developers can merge code changes into a shared mainline where builds and tests then run. Through automated tools the new code's accuracy is verified before integration. With CI your teams can spot and fix security issues and errors easily and much earlier in the development process.

Through automation and regular testing, your teams can iterate faster and increase their overall efficiency. Automatic testing, validating processes and frequently merging changes reduces code conflict. And this is true even when multiple developers work on the same application.

### Continuous delivery (CD)

CD is a software development process typically paired with CI to form a chain of processes for software development, deployment and feedback loops called a CI/CD pipeline. Once code has been tested and built as part of CI, CD is the next part of the process. It runs the final stages to check code is ready to deploy, and can cover everything from provisioning the infrastructure to deploying the application to the testing or production environment. This approach means that development teams produce and

test code in short but continuous cycles. It encourages more incremental updates and enables teams to build, test and reliably release software quickly and at any time.

## **Automated testing**

Automated testing is a process that certifies software is functioning appropriately and meeting requirements before it is released into production. Building it into the CI pipeline is critical for fast and high-quality delivery as it enables your teams to release higher quality code more frequently. With this in place, security issues are detected earlier, enabling your teams to deploy with confidence.

## **Shifting security left**

A shift left approach in DevOps incorporates security into the end-to-end automation. Often referred to as DevSecOps, shifting left integrates testing and the security review process earlier in the software development lifecycle. DevOps integrates security into your CI pipelines, testing code continuously. In doing so, your teams can deal with security issues as and when they arise, as opposed to at the end of the cycle as under the traditional software development model.

When security is left as an afterthought it becomes more difficult to correct issues. It may even be too late to fix them altogether, resulting in delayed deployments, higher costs or vulnerabilities finding their way into production.



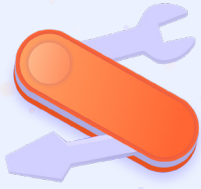
## Documentation

Internal documentation serves as a useful tool when it comes to improving your team's performance and the software you release. Creating and maintaining this documentation improves the services and applications that your team works on and also software delivery and operational performance.

## Feedback

Feedback is an invaluable piece of the puzzle that's essential for improving your overall DevOps process and user experience. A DevOps feedback loop is easy to navigate through rapid iterations and collaboration. It provides access to data that's clear and easy to manage, so feedback is easier to incorporate and deliver improvements. Automating feedback is key, as it ensures quick information collection and distribution amongst the team. Teams can go forward and deliver updates in an efficient manner.





## Other key DevOps technologies you should understand

When discovering all you need to know about DevOps, you can't really grasp it fully without reading up on the technologies that contribute to its success. Key technologies in DevOps range from automation to programming languages to the cloud. Let's investigate these key technologies in more detail.

### The cloud

In terms of DevOps technologies, a good place to start is with the cloud. This infrastructure facilitates scalability, reduces latency and facilitates centralised management. A lot of modern software relies on the cloud and cloud-native infrastructure, and this approach supports easy and seamless collaboration with automated tools and processes. It enables you to deploy code anywhere, faster and more efficiently.

### Collaboration culture

You may have realised by now that collaboration is a big part of DevOps. Through conversation, input and sharing throughout your teams, an open culture of shared responsibility and learning will ensue. Establishing a collaboration culture builds trust among team members and helps to build a better working environment; one where knowledge sharing and information flow freely.



## Automation

Automation plays a big role in streamlining the DevOps process. It's used throughout the development and deployment lifecycle, and is used to perform repetitive, manual tasks consistently and accurately. This cuts down the chance of human error and any associated costs and time spent rectifying mistakes. Although automation doesn't cut out the need for human input entirely, it reduces human contact for recurring tasks such as configuring software environments, integrating code and assisting with CI/CD testing software.

## Monitoring

As your tech stack and team members grow, keeping track of moving parts may feel impossible at times. And that's where monitoring can help. Used by developers, security teams and project leaders, in all stages ranging from planning and development to testing and deployment, monitoring enables constant, real-time awareness of the entire ecosystem.

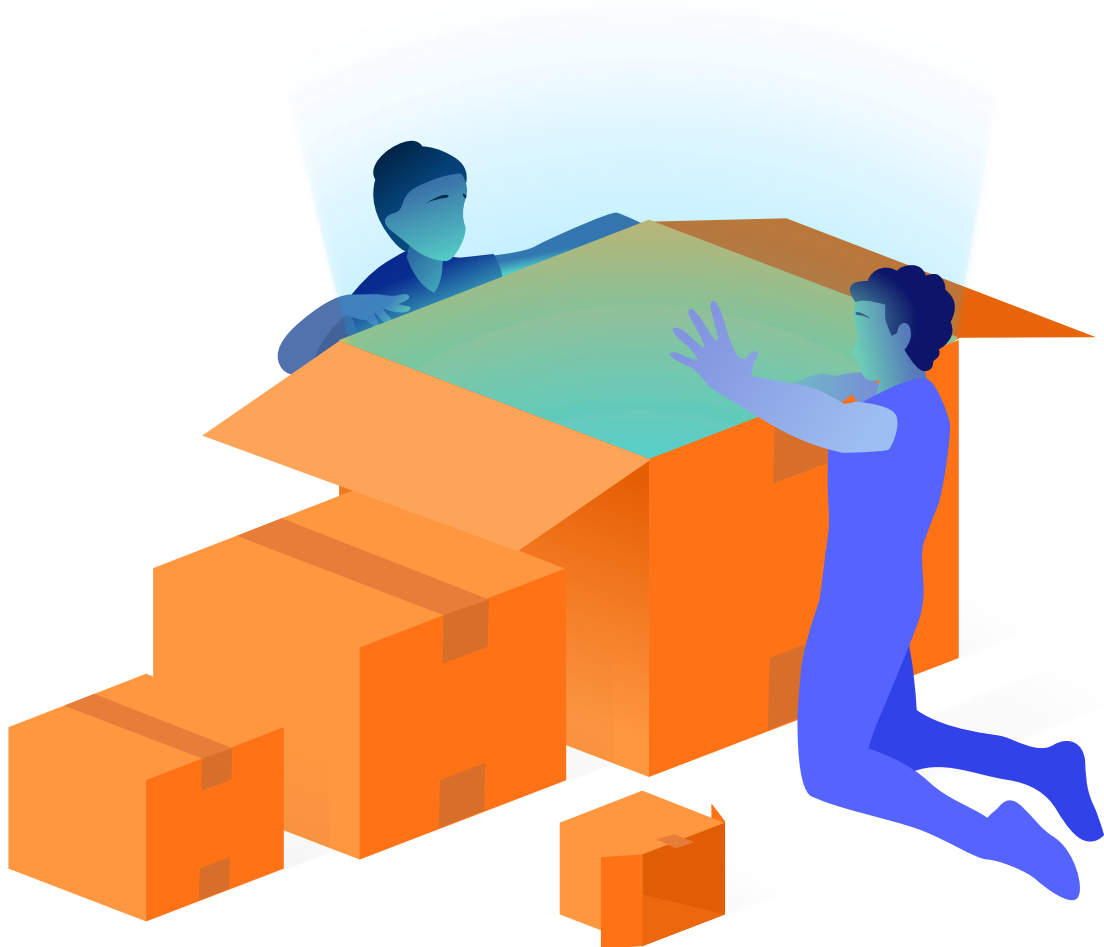
Monitoring keeps track of all moving parts. It keeps track not only of processes, but also alerts any performance issues or threats throughout the pipeline. It helps better detect security issues, increases deployment speed, identifies compliance problems and generally enables teams to work more efficiently.



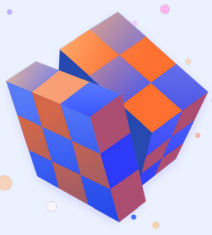
## Containers

Containers allow developers to package everything required to run an app—the code, runtimes and system tools—into one convenient location. This permits developers to create the applications and deploy them on servers with the operating system itself, which makes containers very lightweight and portable.

Containers ensure applications that are built in one environment run consistently and seamlessly in others. This solves the problem of how to get software to run reliably when moved between environments. They are designed to enable your teams to build, test, deploy and maintain applications in a more secure and efficient way, at speed. As they are easily shared between teams, they strengthen collaboration and collaboration culture.







Part four:

## Problems solved by DevOps

As we've already established, DevOps responds to frustrations with the traditional software model and the demand for better software and uninterrupted service. It accelerates the time to market, improves software quality, reduces cycle times and increases customer and internal user satisfaction. Now let's examine how DevOps works in the real-world by looking at some of the problems it can solve.

### **Faster production cycles and increased value for customers**

The constant, growing demand for new features from end users means to stay competitive businesses need to keep up with demand through enhanced and reliable services and products. Therefore, quality and time to market play crucial roles in whether a business is successful in staying competitive.

DevOps accelerates the production process through automation and closes the gap between the two formerly separated teams. With a shorter production cycle and automation of deployment pipelines, DevOps aids team productivity and encourages innovation. Plus, as a result of continuous testing and learning throughout the entire DevOps lifecycle, we see a reduction in errors and better operational efficiency - delivering value faster to customers.

### **Increased productivity equals better customer satisfaction**

Consumer expectation is high in today's fast-paced environment. The expectation is to deliver reliable services and fresh updates in the least amount of time. With that in mind, it's difficult to imagine how spending months testing a new product or feature fits in in a competitive environment.

With DevOps you can test and prepare for production much quicker through automation and deployment of smaller fragments of code. Testing smaller features continuously and making incremental changes to applications helps identify what's working faster and earlier in the cycle, which simplifies the production process and reduces the delivery time. All this means you can respond promptly to internal and end users.



## **Reduced costs through faster time to market**

In principle, through substantial culture change and investment, DevOps will reduce development time and multiply the number of releases per year. By accelerating new features' time to market, companies can be more responsive to their customers' needs and compete on innovation. Furthermore, process automation and continuous testing and improvement reduces downtime and increases security and compliance, resulting in lower operations costs.

## **Improved quality through test automation**

With continuous testing throughout the entire production process, your teams can integrate customer feedback and test smaller fragments of new code alongside the current version. This helps speed up the processes of implementing new features and making security updates to existing products.

## **Improving problem resolution**

At the heart of any successful DevOps practice is close collaboration and a shared responsibility ethos. Better collaboration and communication between your development and operations teams will result in faster problem solving and better code and applications. It will also reduce organisational friction.

DevOps champions the idea of shared ownership among developers and operations teams. It creates a culture that aligns software development with the wider business. A successful DevOps practice accommodates all internal stakeholders to a shared goal, so that everyone is aligned with the goal of bringing value to customers.



Summary:

## Ready to get started?

DevOps transforms software delivery. It enables organisations to build, deliver and improve applications and services at a faster pace, thus unlocking their competitive advantage. Although DevOps is not an overnight fix, it's difficult to ignore the transformational benefits. From improved business agility and enhanced collaboration to reduced costs and better security, the benefits of DevOps impact all corners of an organisation.

If you're ready to level-up, we can help.

If you're ready to get started with DevOps, you're probably wondering about the next stages. The good news is, we're somewhat experienced in this area.

Adaptavist approaches DevOps transformation through intelligent insights and solutions that propel your agile transformation initiatives forward. We provide expert consulting and managed services to deliver end-to-end DevOps solutions, which include building and implementing CI/CD processes and Value Stream Management solutions. And to further strengthen our mission, our partnerships with leading technology partners help us to deliver best-in-class tooling solutions...



That's where GitLab comes in.

GitLab's end-to-end DevOps platform simplifies your workflow by enabling visibility throughout and control over all stages of the lifecycle. It provides an ideal solution for people who are just getting started.

GitLab's DevOps platform draws all DevOps capabilities into one application, so everything is in one place with a unified data store. It means you can avoid context switching and improve user experience, speeding up cycle time.

Through our partnership with GitLab, we deliver end-to-end DevOps services and solutions to help you build and implement CI/CD systems. GitLab is known for industry-leading Source Code Management and CI/CD, so you may get started this way. However, we can help with services that range from DevOps implementation, strategic guidance, assessments, coaching and training, to DevOps as a Service.

[Get in touch](#) to find out how Adaptavist's DevOps solutions, along with our partner solutions, can transform your organisation.



We help organisations transform to continuous change being their business as usual. We do this by supplying technology, providing advice, and delivering change through modern, iterative approaches to development, deployment, and application lifecycle management.

Adaptavist is Atlassian's largest platinum partner, supporting more than half of the Fortune 500. We are uniquely placed to provide our experience, expertise, and insight to help your business.

Whether you want training for your team, to build a software platform for your company, or to automate your existing tooling, we can help you. If you want to unlock the full power of Atlassian and transform your business at scale, get in touch with our team today.

**We're The Adaptavist Group, experts in agile transformation**



Aligned  
Agility

**GRAVITY  
WORKS**



Platinum  
Marketplace Partner

