# DevSecOps: The gateway guide to integral security

The software industry has long needed a more effective way to develop secure products. Cyber-attacks always targeted weaknesses in software, and the rapid digitalization of every aspect of life means that everything from our power supplies to our retail habits are under threat from insecure software.

DevSecOps is transforming the way businesses develop secure products. By ensuring security is part of every step of the development process, the impact of security on costs and timelines is both reduced and made more transparent, whilst overall product security is improved.

## WHAT IS DEVSECOPS?

A combination of development, security, and operations, DevSecOps is a practice that implements security through every step of the software development process.

DevSecOps is an expansion of the practice of DevOps. Whilst security is frequently considered and implemented through DevOps, DevSecOps ensures tangible security decisions throughout the software development life cycle (SDLC). The shift to DevSecOps, from DevOps, entails all the benefits of DevOps – speed, performability and increased flexibility – whilst implementing security in a way that doesn't compromise efficiency.

DevSecOps is a more agile way of approaching security, encouraging more secure outcomes. It's a more streamlined approach as opposed to adding on defence measures after the software program has been built, often at great cost and with significant impact to timelines.

### THE 5 PILLARS OF DEVSECOPS

**CULTURE** – DevSecOps, nor DevOps, are about simply 'add-on' tools or isolated software. DevSecOps is about behaviours and repeated practices, making it an IT culture.

**AUTOMATION** – The backbone of DevSecOps and DevOps is about speeding up deployment. Automation in DevSecOps reinforces faster deployment, utilising technology through each stage of the SDLC to enhance efficiency.

**LEAN** – Lean IT is also considered an organisational culture in IT. It essentially boils down to waste management; eliminating components and processes that do not add value.

**MEASUREMENT** – Software delivery throughput can be measured through metrics with DevSecOps/DevOps. These metrics are valuable to understanding how teams can improve their software delivery.

**SHARE** – Development teams and operational teams have long been divided in their responsibilities and duties. The DevSecOps culture bridges this divide, requiring teams to work together and share responsibilities for operational success.

These 5 pillars make up the CALMS framework. This is something we will talk about in a bit more detail further along in the guide.

## WHAT IS THE DIFFERENCE BETWEEN DEVOPS AND DEVSECOPS?

Essentially, DevSecOps is an evolution of DevOps. Both focus on integration and automation throughout the development process, including incorporating various inputs such as risk management and line of business. The DevSecOps philosophy, however, was born with the goal of deploying security as an unavoidable practice, creating secure infrastructure as code.

### SECURITY THROUGH DEVELOPMENT

Security is never completed, perse. It is something that needs to be developed, repeatedly, and it must be developed within development. With DevOps, it's more common that security will be a looming obstacle at the end of a development cycle, but with DevSecOps, security is deeply rooted in every process.

### CONSISTENT REPORTING

DevSecOps helps to ensure consistent reporting and logging across the board. Unlike DevOps, DevSecOps embeds threat modelling as part of delivery. This way there is a consistency in resolving vulnerabilities and reporting which enhances overall performance.

### REGULATION OF DIFFERENCES

From team to team, each tend to 'do' DevOps differently. So, how can security be effectively practiced or regulated? Without the right framework at the start of a DevOps cycle, you get into a quagmire of change where nobody can truly achieve change because each team is doing it differently.

### SCALABILITY AND FLEXIBILITY

Some may not be able to identify differences within DevOps and DevSecOps. Whilst DevOps is scalable and flexible, DevSecOps is more so. The consistency in standards mentioned through the two previous points offers an increased amount of flexibility. There is an issue of scalability with DevOps. Teams will often be passionate about the culture, adopting it entirely and then realising it doesn't quite scale the way they hoped. DevSecOps enables more scalability.

## UNDERSTANDING COMMON DEVOPS TERMS AND THEIR RELEVANCE IN A DEVSECOPS CONTEXT

DevOps and DevSecOps are more than just a methodology, they are a culture. To truly be able to immerse yourself and your business in the world of DevOps, you must be able to comprehensively understand it. Below are some key terms used in DevOps and how they relate to DevSecOps.

- **INFRASTRUCTURE AS CODE** – This is a concept where, when applied to DevOps, manages infrastructure code in the same way that code for any other kind of application would be managed. As opposed to manually adjusting code, infrastructure as code (IaC) can be maintained through development. This is key in both DevOps and DevSecOps and is used for version control, continuous monitoring and continuous delivery.

- **CI/CD** – Continuous integration (CI) is a practice where developers frequently integrate their code changes into a central repository. These code changes will then go through an automated build and test run. Continuous delivery (CD), sometimes referred to as continuous deployment, is an expansion of CI, where code changes are deployed to a testing environment, following the building stage. CD integrates all changes, such as bug fixes and experiments, into production. Using CI and CD makes for good practice for developers and security teams. These are processed through short cycles, allowing for faster and more frequent software delivery.

- **PIPELINE** – often referred to in the context of a continuous delivery pipeline, a pipeline is a set of processes your code changes will undertake on their way to production. Automated tests, builds and deployments are coordinated as one workflow. Pipelines in reference to DevSecOps simply means integrating security as an ongoing process through your DevOps pipeline.

- **AUTOMATION** – Whilst automation isn't exclusively a DevOps term, it does go hand in hand with the practices of DevOps. Through the DevOps lifecycle, there is a wide variety of processes that take place. Automation, conducted by technology, enables these processes to be streamlined, fast and free from human error. In the context of DevSecOps, incorporating security principles through automation allows for shorter feedback loops through the SDLC which enables engineers to identify and resolve security issues at a rapid rate. Automation as a part of DevSecOps empowers teams, accelerating security and creating a seamless operation for businesses.

## THE BENEFITS OF A DEVSECOPS APPROACH?

The overriding benefit of DevSecOps, in a sentence, proactive, end-to-end security, integrated though each phase of the software development cycle. This practice has been introduced to accommodate the ever-changing nature of the IT landscape and can hugely benefit all kinds of organisations involved in developing and deploying software applications.

Placing security earlier in the development process is one thing, but this is about embodying a culture of secure first code, continually using metrics to spot trends, sharing all findings across the organisation, automating everything and driving lean methodologies to create highly efficient delivery teams.

**COST-EFFECTIVE** – An EMA report conducted in 2017 found that DevSecOps initiatives had better ROI (Return on investment) over traditional security infrastructures. The report also found that SecOps best practice reduced operational costs.

DevSecOps can help reduce costs by integrating security into code from the very beginning. It is often costly to repeat a process in an attempt to fix a security issue that teams come across. With the delivery of DevSecOps, organisations can avoid security issues that happen further down the line, which can result in time-consuming rebuilds and the late release of products.

**THREAT MODELLING** – Threat modelling is a key component in DevSecOps. In most DevOps practices, threat modelling tends to get neglected by developers. Maintaining the DevOps ethos of keeping code short through each step of the SDLC allows for a consistent assessment of vulnerabilities in an application in DevSecOps.

**SPEED** – DevSecOps manages to imbed security through code, without compromising on deployment time. Code is shorter in DevSecOps, which creates digestible chunks of code that can be managed and changed, quickly and easily.

**AGILITY** – DevSecOps maintains, if not increase, agility for businesses across operations. Release cycles are shortened, enhancing cybersecurity and overall delivery. Continuous delivery, integration, logs, monitoring and scalability are treated through the agility that DevSecOps offers.

The ethos of embedding security throughout the entire SDLC, preventing other time-consuming issues at the end of the process, also enables a degree of agility for teams.

**SECURITY AND DEVELOPMENT TEAMS COLLABORATE MORE EFFECTIVELY** – DevSecOps encourages development and security teams within an organisation to work together, combining skills and knowledge to solve problems. The SDLC is a process that, whilst predominantly in the hands of developers, is enhanced and optimised through other practices such as implementing security.

This security practice requires teams to collaborate to ensure alignment across the application lifecycle. As teams tend to work together in DevSecOps, as opposed to working in silos, operations function more smoothly, with the chance for teams to effectively communicate and share ideas.

**IMPROVED SECURITY** – DevSecOps encourages security considerations and prompts teams to understand crucial principles and their organisation's security posture.

Throughout the entire SDLC, code is audited, reviewed and scanned to test for security issues. Like anything we 'practice', this is done repeatedly to enforce optimal conduct. If security issues are detected, they are dealt with before other dependencies are introduced.

DevSecOps solutions frequently remind teams of their security responsibilities. Through collaboration, making it everyone's responsibility, security implementations are made stronger.

**MORE EFFICIENT DEVELOPMENT AND DELIVERY** – DevSecOps irons out the development process, setting a steady foundation for code through every stage. As a result, the application is stronger and more reliable than one that hasn't been built through DevSecOps. By identifying security issues early in the process, these can be effectively solved, and development can adapt and evolve accordingly. DevSecOps helps with noticing certain repetitive security issues. These can then be addressed with the right training or tools, which isn't just good practice, but, through the learning process, can provide valuable knowledge to relevant teams.

## COMMON DEVSECOPS SECURITY TERMS: A BRIEF EXPLANATION

**Below are some security terms that are used in the DevSecOps culture. Looking through these terms will help to deepen understanding of DevSecOps and the various processes involved in the practice.**

SAST – An acronym for Static Application Security Testing, SAST is where static and precompiled code undergoes a scanning process to discover potential vulnerabilities.

DAST – Dynamic Application Security Testing is similar to a penetration testing tool. Through simulating an attack on a system, DAST tools can find vulnerabilities in a running application.

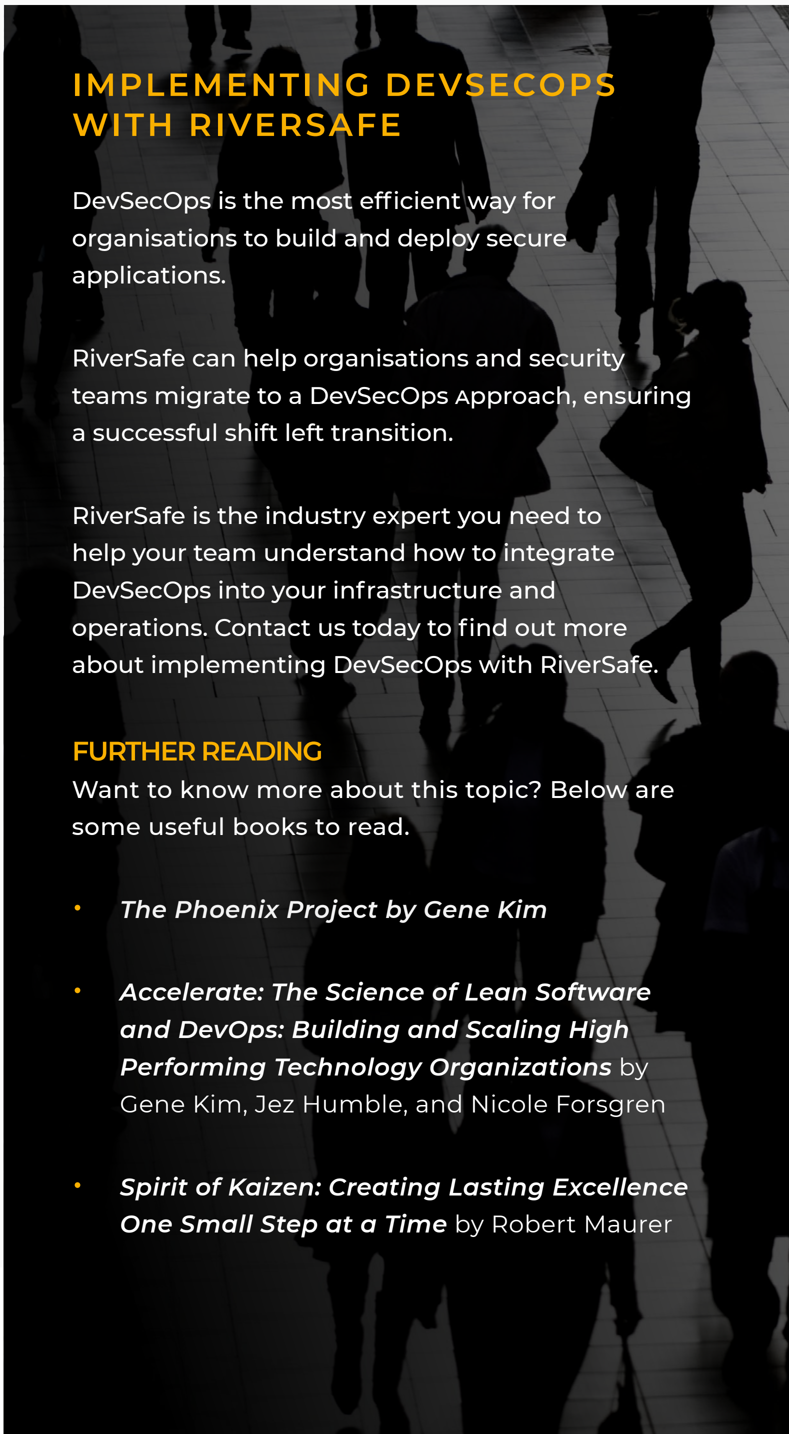
SBOM – Software Bill of Materials informs teams about what open-source software components are being used and identifies the architectural or license risks and security vulnerabilities in those components.

SECRET SCANNING – Secret scanning tools scan repositories for known types of digital credentials to prevent deceptive use of accidentally committed secrets.

CALMS – As discussed earlier, Culture, Automation, Lean, Measure and Sharing make up CALMS. CALMS is a model that assesses an organisation's DevSecOps structure. Because DevOps cultures are evolving, maintaining regular evaluations through models like CALMS helps to ensure teams and systems are maturing with the practice.

FUZZING – This is an automated software testing technique which, in its essence, sends intentionally invalid data to a product with the aim of activating an error condition. These triggered error conditions can then signify system vulnerabilities.

SHIFT LEFT – Shift left encompasses the entire DevSecOps ethos. It encourages software engineers to shift security from the right of the SDLC to the left –the beginning of the delivery process. Security is typically done through the 'right-hand side', which is when it's implemented at the end. This right-hand approach can result in significant work at the end of the development process.



## IMPLEMENTING DEVSECOPS WITH RIVERSAFE

DevSecOps is the most efficient way for organisations to build and deploy secure applications.

RiverSafe can help organisations and security teams migrate to a DevSecOps Approach, ensuring a successful shift left transition.

RiverSafe is the industry expert you need to help your team understand how to integrate DevSecOps into your infrastructure and operations. Contact us today to find out more about implementing DevSecOps with RiverSafe.

### FURTHER READING

Want to know more about this topic? Below are some useful books to read.

- ***The Phoenix Project by Gene Kim***
- ***Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*** by Gene Kim, Jez Humble, and Nicole Forsgren
- ***Spirit of Kaizen: Creating Lasting Excellence One Small Step at a Time*** by Robert Maurer