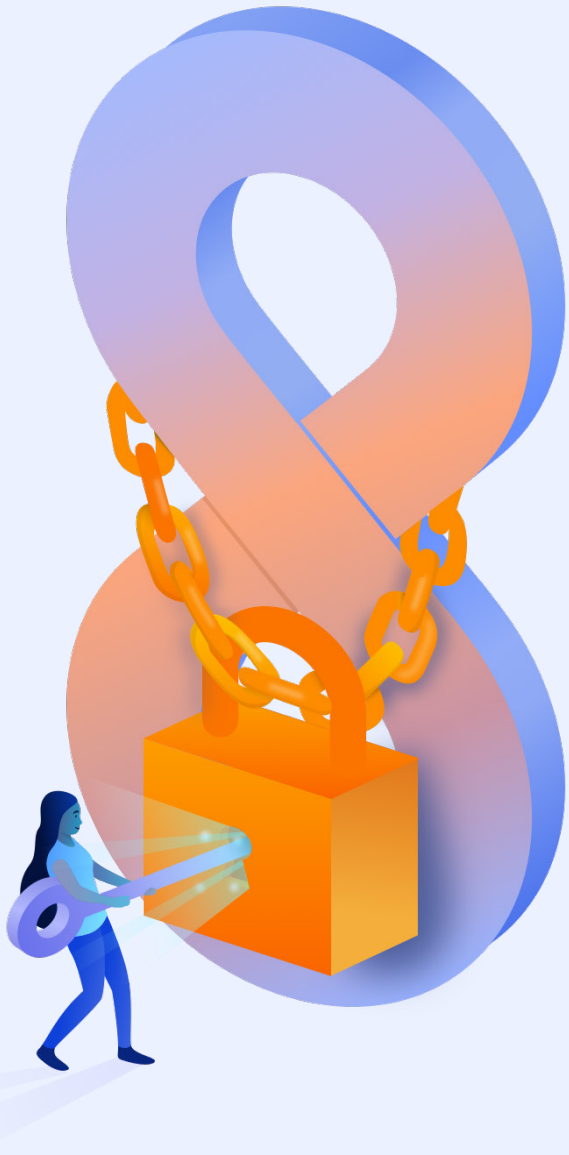


The enterprise guide to unlocking DevSecOps' fullest potential

Adaptavist

OKRs vs product roadmaps: the ultimate guide

How OKRs define the product roadmap

Contents

Introduction

It's time to step up DevSecOps 1

Part one

A security-first approach 3

Why are we doing DevSecOps again? 4

So what's next? 5

Part two

Bring on best practice 7

People 8

Process 9

Tools 10

Part three

Next-level DevSecOps 11

Strive for excellence 11

Take a look at your tools 12

Embrace application security testing 13

Connect with the value you're creating 13

Make more of machines 13

Be open about open-source 13

Find the problem and fix it fast 14

Bring others on board 14

Create shared goals 14

Conclusion

Unlock DevSecOps' full potential 15

Introduction:

It’s time to step up DevSecOps

As a large enterprise with a security-centric outlook, you already have established security teams, practices, and tools in place. But with security excellence as your goal, and the goalposts changing all the time, this is not the time for complacency.

When it comes to your DevSecOps strategy, there’s always room for improvement, wherever you are on your journey and whatever tools you use.

This guide focuses on what large companies like yours need to consider to ensure you’re getting the most out of doing DevSecOps – whether that’s upskilling your people, powering up your processes, or getting more bang for your tooling buck.

In this guide, we cover the following

- Why DevSecOps is so important for an enterprise like yours.
- Best practice considerations to help you stay on top of DevSecOps.
- Expert ideas to level up the way you do DevSecOps.

DevSecOps stands for development, security, and operations. It is a practice of integrating security into all stages of the CI/CD pipeline. It is also about having the mindset of security and best practices when we’re performing our duties.

Traditionally, security checks would take place at the final stages of the CI/CD, which could be costly and time consuming, especially if it meant reworking a huge amount of code when a threat was found at this late stage. A DevSecOps approach embeds security right from the start, helping to identify issues much earlier.

We're also seeing an increased adoption of GitOps practices, ensuring that DevSecOps can scale effectively across teams and environments. By leveraging Git as the single source of truth for security policies and infrastructure configurations, GitOps enhances security automation, ensures version-controlled compliance, and reduces manual intervention. By integrating security as code, organisations can proactively detect vulnerabilities, implement policy-driven controls, and maintain a more resilient security posture.

DevSecOps impacts every part of your organisation for a complete transformation of how you approach and implement security. But wholly embracing these practices is an ongoing process that requires expertise and support.

Suppose you're interested in boosting your DevSecOps implementation but aren't sure how; this guide is a great place to start. And our team of experts is here to help when you're ready.

Part one:

A security-first approach

Your enterprise is no stranger to DevSecOps. Whether you're focused on an application, IaC, or pipeline security (or all three), you know that not prioritising security has huge drawbacks – costly delays and inadequate short-term patching being just a couple. That's why you've already applied security policy and technology to DevOps, transforming your software development life cycle and embedding security more deeply across your organisation.

As a result, you should be able to identify security issues much earlier. With security testing, monitoring, and reporting part of the CI/CD pipeline, security standards should have become hard-wired into your infrastructure, and fast feedback loops should be helping you stay on top of security and remediation.

But you've also undoubtedly learned that DevSecOps is not a one-size-fits-all solution or a mere box-ticking exercise. Your approach will be unique, influencing your culture and how you automate, design, and develop your software. And it requires an organisational mindset shift, where security becomes everyone's concern

Why are we doing DevSecOps again?

When you get in a car, you put on the seatbelt without even thinking. You know it's designed to keep you safe, but you don't think too much about why or how. And with DevSecOps being part of your approach for a while, you would be forgiven for forgetting why it's so important.

You must have a robust cloud computing platform with flexible data and storage solutions as an enterprise. But software development also demands fail-safe security and compliance to combat the threat from hackers. According to the Global Cybersecurity Outlook 2024, there have been increasingly alarming attacks against critical infrastructure and elements in global supply chains. Not to mention, the rise in AI used in cyberattacks, with risks including the use of deep fakes, misinformation, and algorithmic manipulation of social media.

An ever-present threat

In 2024 the World Economic Forum's Global Risks Report listed AI-generated misinformation and disinformation, and cyberattacks, alongside extreme weather, societal and/or political polarisation, and the cost-of-living crisis as the risks most likely to present a material crisis on a global scale.

Cybercrime is a problem that's not going away. DevSecOps is the defence enterprises need to stand a chance against these criminals. If attacks aren't picked up early and dealt with immediately, you risk releasing products or features with viruses, malware, and other security risks.

And the results can be catastrophic. From abuse of your organisation's intellectual property, loss of revenue, and unforeseen costs relating to the breach to serious knock-on implications for your customers, your reputation, and ultimately your organisation's success.

DevSecOps can have a direct impact, helping to manage these challenges and prevent incidents from occurring in the first place. It ensures your

enterprise documents and implements all necessary security requirements, incorporating security into design, development, and testing. And if you're making any changes, you think about security first. Ultimately, your developers will code more securely, and you'll build trust with your customers.

So what's next?

Right now, you've made real progress and are already reaping many of the benefits. From increased sales, lower costs, and faster delivery to improved responsiveness, easier compliance, better collaboration, and more flexibility, you are hopefully tapping into plenty of big wins from switching up your approach to security.

But implementing DevSecOps successfully is not as simple as flicking a switch. It's a continual assessment and improvement process to ensure you're getting the most out of your processes and tools – and to ensure a 'security-as-code' culture is embedded across your whole enterprise.

We get it. You've done a lot of the hard work and have DevSecOps teams, practices, and tools in place. And you've noticed significant improvements – from culture to cost-cutting. But could it be better?

Ask yourself the following questions:

- Are our response strategies efficient enough?
- Are we always meeting industry regulations?
- Are we struggling to eliminate security bottlenecks?
- Could our incident-recovery rate be faster?
- Are collaborative teamwork and cross-team communication a bit hit or miss?
- Are we automating enough to free people up to focus on higher-value work?
- Are we keeping on top of cybercrime innovations as well as we would like?

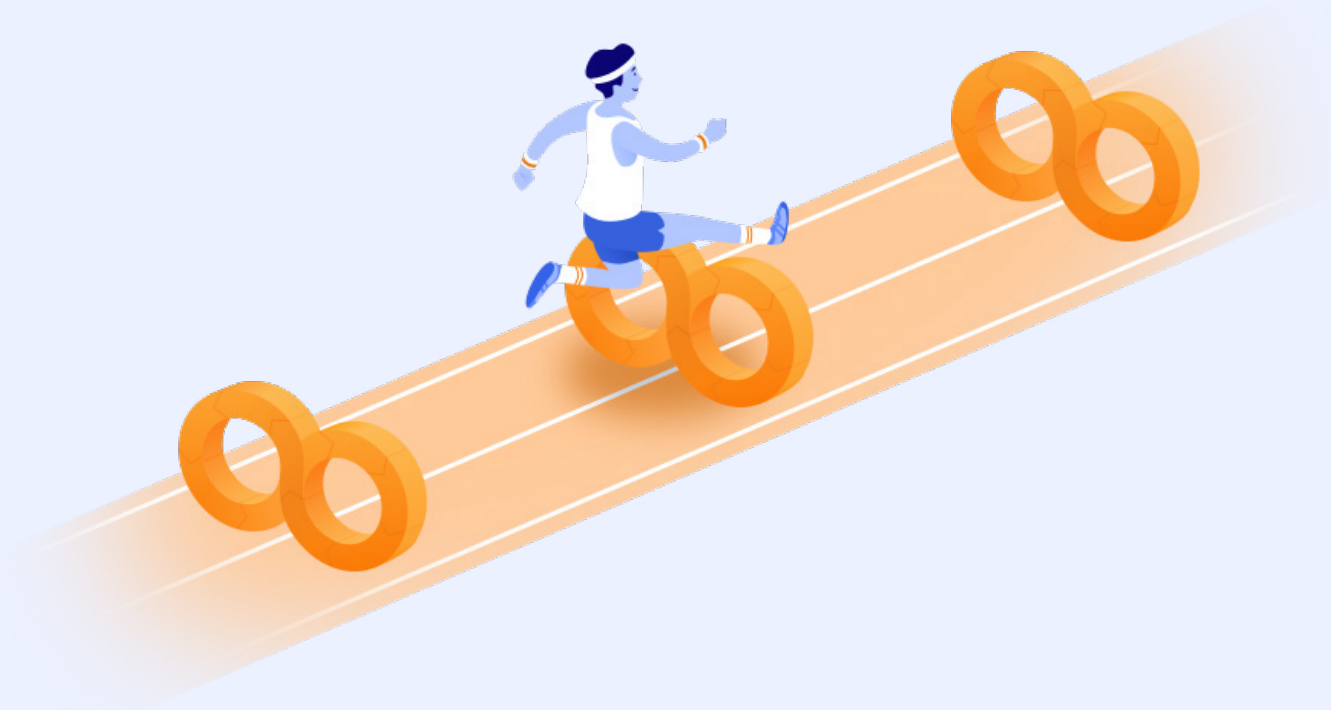
If you're not 100 percent happy with your answers, there's definitely room for improvement. Whether that's realigning around best practice, which we'll take a look at next, or taking some new steps to boost your DevSecOps implementation (see page 11), there's plenty you can do to make DevSecOps work harder for your enterprise.

Part two:

Bring on best practice

Like any digital transformation, a successful DevSecOps implementation requires you to address three big components: people, processes, and tools with security at their core.

You’re already well on your way to reframing organisational thinking and making your software as safe as possible, but refreshing your entire enterprise on best practice will ensure you stay on top of DevSecOps and continue to embed this thinking moving forward.



People

Your people are essential to DevSecOps success. You might already have faced significant challenges supporting people to shift from a traditional software development approach to putting security first. But you need to keep at it – it's buy-in from your people that will ultimately ensure DevSecOps continues to flourish. When it comes to your people, here are three key areas to focus on.

Collaboration

Keep your eyes open for siloes working, where developers, operations, and security professionals see each other as problematic rather than all being on the same team. Siloes will stop you from scaling and speeding up. And poor communication will result in duplicated work.

> If you haven't already, think about installing a security champion as part of a central tooling and enablement team. They would liaise with the sec team to emphasise the importance of security and ensure security interests are baked in from the start. This can make a huge difference, helping foster cross-team communication centred around security

Skills

If skills are in short supply, don't throw your hands up in defeat. There are lots you can do to have an impact. Ensure you're investing in your people with good quality training rooted in your security goals. And don't just focus on new hires – existing staff deserve this attention too. Training should be rooted in your enterprise's standards and security goals and flexible and tailored to specific teams and their needs.

Culture

Have you seen a change in your culture? Have people adjusted their attitudes around security? Everyone should feel a responsibility to ensure security is front and centre when decisions get made – and there shouldn't be a single security team any more than everyone sees as a blocker to getting work done

Process

The common processes that should be built into your organisation include automation, shifting left (and everywhere), and maintaining strict coding standards.

Automation

This helps you to introduce controls and testing early and often, ensure compliance and implement effective version control to aid recovery.
For example, are you creating compliance requirement metadata and incorporating it into your assets? If you're pushing code to production frequently and not automating enough already, you might be compromising security to meet the demands of code delivery.

Shifting everywhere

An evolution of the above, this practice emphasises runtime protection and continuous feedback loops from production environments. It relies on runtime application self-protection (RASP) tools and observability platforms like Datadog or New Relic. These actively monitor and protect your application while it's running, detecting and blocking any malicious activity in real-time.

Strict coding standards

If you're setting and maintaining strict coding standards, you'll spot issues much earlier. When you make changes to code, even small ones, are these being checked against your recommendations? It's also vital that those standards are regularly assessed. That way you can consider any new security threats or needs.

Tools

There are plenty of tools out there claiming cure-all solutions. On the whole they're there to carry out scans, audits and testing throughout the SDLC to ensure security is built-in rather than applied retroactively. Largely, they fall into four main types:

SAST

Static application security testing, is a mature tool that scans the source code repository to identify vulnerabilities at static file time. While scans can take a long time, it allows for early detection and can scale easily with your development team.

DAST

Dynamic application scanning tools scan staging and production sites to detect live application flows, such as user authentication and SQL injection. These are easy-to-use tools that test your code at runtime and allow for simple validation.

SCA

Software composition analysis lets teams identify, track, analyse, and in some cases remediate any open-source code that's added to a project, scanning dependencies for vulnerabilities. SCA also checks for which open-source licences are being used, helping to mitigate legal risk.

IAST

Interaction application security testing (IAST) is a more recent technology that mixes SAST and DAST to provide results in real-time. Ideal for QA testing, these tools integrate seamlessly into your CI/CD pipeline.

Part three:

Next-level DevSecOps

So you're following best practices, and things are moving along nicely, but what will it take to really step things up and reap more from your DevSecOps efforts? Here are a few ideas to take things up a notch.

Strive for excellence

Across the business, DevSecOps adoption will be at various stages. Knowledge, staffing, skills, and training will differ, as well as attitudes and experiences with DevSecOps. By establishing a Centre of Excellence, you can create a cross-functional team of experts from different parts of the organisation with a united goal – improving DevSecOps adoption.

Take a look at your tools

New tools are constantly being released to the market, so it's important to ensure your DevSecOps stack includes the best and most up-to-date technology. A couple of vendors we're proud to partner with that are transforming DevSecOps practices for our customers are GitLab and Sonatype.

GitLab
GitLab is a complete DevOps platform that covers all stages of the DevOps life cycle. It incorporates value stream reporting, planning tools, CI/CD and more. It lets you automate your security and compliance policies within the CI/CD pipeline, so you'll know if anyone changes anything anywhere. And it was designed for everyone who cares about your code to use, from devs and ops to your security professionals.
Find out more about GitLab

Sonatype
This developer-friendly software supply chain management platform helps accelerate innovation while improving application security. Powered by the Nexus Platform, it analyses over 100 million open-source components, feeding its results to users to eliminate the friction of manual governance so that they can make better decisions across their SDLC.
Find out more about Sonatype

> Once an application or its supporting infrastructure has been deployed, the process of DevSecOps does not stop. Companies should continue to monitor and observe the application in the production environment. Runtime application self-protection (RASP) continuously monitors inputs and filters out the negative ones that could allow attacks. These tools also have the ability to stop these attacks on run time.

Embrace application security testing

It's not good enough for application testing to be just something you tick off a list as it goes to production. Continuous integration is essential for speed and quality, rather than QA teams testing builds that are days or weeks old, not knowing which build defects belong to. Reassess what happens when vulnerabilities are found too. How severe does a vulnerability need to be to stop the pipeline? And what's the process for vulnerability feedback getting back to dev teams?

Connect with the value you're creating

A project mindset, where you're focused on speed and deadlines and achieving individual goals, won't suffice in this age of agile software development. You need a consolidated view of what value each team is delivering and how it all joins up.

Make more of machines

Great communication can make all the difference in ensuring DevSecOps doesn't flounder. Talk to your teams about the real issues they're facing. SAST can mean teams spend a lot of time chasing false positives – this can increase the chance they'll miss something important.

Be open about open-source

Your developers are definitely using open-source tools, but how are you protecting your own assets in the process? Ask the tough questions about how open-source tools are used and maintained, how reliable your software inventory is, and whether code from third-party vendors is validated.

Licence compliance features found in tools available from Sonatype, GitLab, and Snyk can be life-savers here, letting you search your project's dependencies and decide whether to allow or deny licence use accordingly.

As your system gets more and more complex, it's vital that you have best practices in place to protect your software supply chain. That includes considering emerging standards, like the Open Source Security Foundation's initiatives, which include Supply-chain Levels for Software Artifacts (SLSA). This checklist of standards and controls is designed to "prevent tampering, improve integrity, and secure packages and infrastructure".

Find the problem and fix it fast

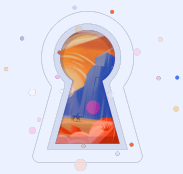
It sounds simple, but if you can speed up your process, finding and fixing problems before someone else has the chance to exploit them, you'll maintain more trust with your customers. A lot of this depends on your tools, so make sure they have the capabilities you need to identify blind spots and quickly cross-reference new issues with your own software.

Bring others on board

If DevSecOps is already making a big difference to your developers, operations and security teams, don't be shy – keep other departments informed. You might want to educate finance staff about subscription-based DevSecOps tools or loop in legal so that they can weigh in on potential open-source licensing concerns.

Create shared goals

Most initiatives are more successful when everyone is rowing in the same direction. If you're not already doing it, bring DevOps and security teams together to set goals, treat them as one team, and settle any conflicts up front. Centralise any KPI or OKR reporting with the help of a tool like Jira Align so that everyone can see where the goalposts are, how much progress has been made, and how individual and team-level goals relate to the wider organisation.



[Conclusion]

Unlock DevSecOps' full potential

With high-profile cyber attacks occurring with alarming frequency, customers are prioritising security more than ever before – it's become a highly prized commodity in its own right.

While there is undeniably incredible pressure to move fast and release features, particularly in the face of stiff competition, you build more credibility into your products when you incorporate security at the planning and design stage and identify vulnerabilities before production rather than when it's already too late.

As enterprises move towards hybrid and multi-cloud environments, we're seeing a rise in Zero Trust Architecture (ZTA) as part of modern DevSecOps. It's based on an evolving set of principles that focuses on users, assets, and resources. This approach enhances your security posture by eliminating implicit trust, enforcing strict verification across networks, ensuring least privilege access, and reducing surface attacks.

Following DevSecOps best practices at every stage is crucial and the first step in defending your products. It's important to master the basics before layering on more complexity. But pushing your DevSecOps implementation further can truly set your products apart from your competitors. And it's something we're proud to help enterprises accomplish every day.

Wherever you're at in your DevSecOps journey, we're here to help.

Adaptavist combines partnerships with leading technology providers and expert consultancy to deliver DevSecOps solutions tailored for your people, products, and end-users. From maturity assessment to see where you're at to training, strategy, tool implementation, and integrations, we have the experience and knowledge to take your DevSecOps efforts to the next level.

Want to know more about how we can support your digital transformation success?

Get in touch

Platinum Solution Partner US GOVERNMENT

aws PARTNER Solution Provider

monday.com certified partner

GitLab Select Partner