ATLASSIAN

Bridging the gap

Three modern frameworks to modernize the ways your teams work

Table of contents

Executive summary

Organizations across the globe have had to rethink how they can meet their customer's growing demand. Many have answered the call by prioritizing digital technologies. But technology alone doesn't enable you to deliver value faster.

For technologies to be beneficial to your organization, you need to look at how your teams are working. Traditional development practices that have driven success in the past don't work quite as well at scale.

To meet the needs of your customers, you'll have to also align your teams and build practices that put your customers front-and-center.

The three frameworks that are going to help you move more seamlessly as an organization are:

DevOps

DevOps bridges the gap between development and operations, where both teams are committed to building quality code and delivering it quickly to customers.

SecOps

SecOps takes security from being owned and managed solely by your security team and makes security everyone's priority.

DevSecOps

DevSecOps brings both frameworks together to focus on delivering code efficiently and securely.

Once the practices and processes are in place, the products can come in and further support your needs. For organizations operating in a self-managed environment, Atlassian Data Center products have the advanced capabilities you need to meet your customers needs at scale.

In this whitepaper, we'll cover how to build these frameworks into your organization's ways of work and how you can use Data Center to support your needs.

What changed?

People are relying on their products and apps more than ever before. Whether they're using online banking, engaging with a government website, or just ordering takeout, these modern-day conveniences have led people to engage with technology in ways they never have before – and at a much faster rate.

And as an organization that is building these products and apps, you have to balance meeting growing customer demand with supporting your internal teams' evolving needs.

But all too often, as demand increases, your development and operational processes can't withstand the additional pressure. Traditional ways of development aren't conducive to delivering value to your customers as quickly as you need to.

Additionally, the need for secure software is increasingly difficult to ignore, with a growing number of cybersecurity threats leaving you vulnerable to a breach. Traditional operating models no longer suffice – organizations need a scalable, defined approach to secure software delivery.

Now, leadership teams are turning to admins, like you, to help make decisions on how to improve ways of work throughout the organization and drive your organization's digital transformation strategy.

01

Challenges your development teams face

Challenges your development teams face

Any enterprise admin will tell you that scaling quickly often comes with organizational baggage. Development teams tend to build practices and processes based on their needs at a given point in time, but those needs change as organizations grow and mature and work becomes more interconnected.

Here are some examples of organizational baggage and how it may be impacting your development teams:

Team responsibilities don't match organizational structure

Many organizations have adjusted their reporting strategy and business objectives to support more cross-functional collaboration. However, at the team level, it's not uncommon for teams to continue to work in silos.

Historically, teams have focused on their own objectives, whether that's releasing a new product, delivering a feature by a certain date, or building more automated tests. Now, your organization has its own overarching goals, which means that all of your team's work needs to contribute to the bigger picture. However, silos between your teams can limit collaboration and impact how you deliver on those objectives.

Part of this is that roles and responsibilities change at scale – they have to – but all too often, this change causes confusion amongst teams. Who's responsible for optimizing security in a product? Is it the product team, or is it the security team?

When teams lack clarity around the different tasks that they are responsible for, things start to slip through the cracks.

Tools are inconsistent or outdated

Teams adopt tools and products that help them get work done faster, but they don't always choose the same products that other teams are using.

For IT, it's difficult to support a growing tool chain. You need to ensure that the products your teams use are maintained and secure, which is difficult if these products don't go through your organization's procurement process.

Plus, these products may not be capable of integrating with each other. So, if teams can't communicate with each other through their products, they can't collaborate efficiently.

Releases take too long

The success of your organization depends on you delivering value and, since we live in the age of instant gratification, customers expect that they are getting value frequently and consistently.

Developers often don't check in their code until it's completely done. As a developer, if you're only focused on building a specific feature, this makes sense. You wouldn't want something to be pushed live until you've finished building it completely. However, by not checking in code regularly, you increase the risk of slowing down the development process.

Waiting to check in code until it's done may conflict with changes that other members of the team have committed. Builds then start to fail because the code isn't compatible; and because it wasn't checked in incrementally, the effort to remediate this wider diverge is considerably more difficult and time-consuming.

This situation also impacts your quality engineering (QE) team. When they finally get access to a build for testing, if something isn't working as expected, pinpointing exactly what caused the bug is difficult.

Code quality and security is compromised

To get value out the door, your development teams are working overtime developing new features, as quickly as they can. The need to build faster often comes at the cost of two things:

- Quality
- Security

Time creates pressure and your development teams are up against the clock. So teams may not be given the necessary time to test comprehensively or to ensure the level of confidence desired.

Security teams face similar pressure. For example, they may identify a security fix, but making the fix would delay the release. So, the fix is pushed, but the next release has an aggressive deadline too, so these fixes are deprioritized.

Manual tasks are bogging down your teams

Even though you may have added more people to your team, you're still faced with too many manual tasks. In fact, the number of tasks that you need to do manually has probably grown even larger.

Many teams are still bogged down manually completing all of their day-to-day tasks – even though these tasks, such as running QA tests or kicking off a build after code has been committed, are relatively simple tasks that could be automated.

So, how can organizations overcome these challenges?

Build a DevOps framework

To move past these challenges, you need to adjust your practices and processes. That's where DevOps comes into play.

DevOps is a set of practices you can use to bring your software development and IT teams closer together by aligning them with the shared goal of delivering value to your customers efficiently and effectively. Rather than each team focusing on a small piece of the development process, IT and development work together to build, test, and release code.

Building this type of strategy requires three things:

People

Processes

Products

THE BENEFITS OF APPLYING DEVOPS IN YOUR ORGANIZATION:

- Align teams to deliver value to your customers faster
- Gather customer feedback directly and build it
- Build collaboration and trust across your teams
- Manage work and resources better

Get team buy-in

For DevOps to be successful in your organization, first and foremost, your teams have to believe in its value. You can have the best products and practices in place, but if you don't have people in your organization championing your DevOps strategy or teams willing to change the way that they work to fit this new framework, it won't be successful.

Tips for creating a DevOps culture

- ☐ **Practice empathy:** Understand what each of your team's challenges are so that you can work together to overcome them. People learn empathy by having similar experiences, so allow your dev teams to be on call to understand what operations teams face and vice versa.
- ☐ **Change incentives:** Work with leadership to align on the goals of your teams so that everyone is working towards the same thing.
- ☐ **Update success metrics:** Identify a shared goal, such as optimizing your customers' experience on your website, and have your teams focus on that. This will help your teams think outside of their traditional roles. Dev teams will be more empowered to fix stability issues, and your IT team will be more open to innovating.
- ☐ **Reframe failure as part of the growth process:** Allow your teams to experiment as you move toward a continuous delivery model. There may be some failures, but you'll be able to learn from the experiences and ultimately find better ways to deliver value.

Changing the way that teams work can be difficult. Take the shift to remote work as an example. Most people had to change the way that they interacted with their teams – learning how to communicate information effectively while navigating the all too real meeting fatigue. But we overcame.

And much like we needed to train ourselves then, you'll want to train your teams in DevOps principles to reduce friction and unite everyone in your organization around your shared strategy.

What about my non-technical teams?

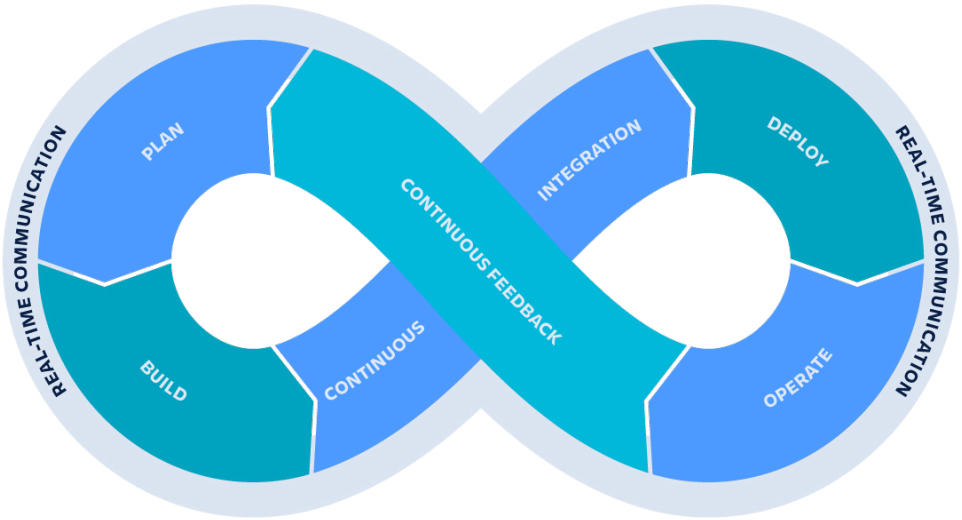
One of the biggest challenges that many organizations experience is resistance from non-technical teams as they believe DevOps does not work for them. That isn't the case. It's just going to look different.

For example, let's say you have a team dedicated to creating training material for your organization. Are they going to be checking in code regularly and have artifacts built continuously? Probably not, but the other phases still apply. They'll still want to prioritize the scope of their project, build the modules out and get feedback before rolling it out to production, and get feedback from customers and internal teams who have taken the training to inform the changes they want to make in the future. In essence, your team is applying DevOps practices in a way that works for their team to deliver value faster.

DevOps phases

Traditional software development follows a rigid, linear model, which is different from DevOps. DevOps practices are structured more like a loop where teams think about software development as a continuous cycle versus a line with a start and endpoint. The DevOps lifecycle consists of six phases:, representing the processes, capabilities, and tools needed for development on the left side of the loop, and the processes, capabilities, and tools needed for operations on the right side of the loop. Throughout each phase, teams collaborate and communicate to maintain alignment, velocity, and quality. The DevOps lifecycle includes phases to plan, build, continuously integrate and deploy (CI/CD), monitor, operate, and respond to continuous feedback.

Let's look at each of the phases in detail.



1

PLAN

For any release to be successful, you need to focus on the most important initiatives first and understand the work involved. Without all of your teams aligning on timelines and resources, you run the risk of missing your release deadlines.

Teams often turn to software planning tools and adopt agile methodologies to help them plan their releases. Agile methodologies are sets of practices that help you and your teams breakdown releases into manageable tasks and milestones. Agile relies on sprints, backlogs, epics, and stories to assign work to skilled team members, adjust timelines when necessary, and deliver quality products and services to customers.

Planning isn't just relegated to deciding what features you're going to build in a particular release. Planning tools become an increasingly important part of day-to-day communication throughout the software development process between both technical and non-technical teams. By providing teams with a centralized place to plan and track their work, your organization can quickly get updates on releases.

2

BUILD

During the build phase, you create your code.Based on what was prioritized during the Plan phase, your development teams can do what they do best and build their code.

3

CONTINUOUSLY INTEGRATE AND DELIVER (CI/CD)

CI is the principle in which developers check in their code to a central repository on a continual basis. This enables developers to integrate smaller changes, receive and act on feedback sooner, and reduce the amount of time it takes to validate it, all of which ultimately improves the quality of their code.

Checking in code more regularly also requires less effort integrating their code into their repository and complete security scans sooner.

It's important to call out that agile and DevOps aren't at odds with each other. In fact, to fully deliver on your goals, you need to integrate both into your software development process. DevOps can help you think more strategically about the code you release, and agile will help you break down those tasks to help you prioritize your work and provide more transparency between your teams.

Highly mature teams with test and release discipline may take their CI/CD practices a step further and let their code flow into production without human intervention.

But the story isn't done there. Once you've set up the CI part of your pipeline, you need to add continuous delivery (CD). CD is the practice of code changes being automatically prepared for production. Essentially, all of your code is built and deployed to a staging or production environment. And, because you've already run the code through your automated test suite, your code should be bundled in a deployment ready artifact.

4

MONITOR

Historically, operations has been siloed from development, which means that development has been responsible for running essentially anything that they build. However, this doesn't fit with a customer-first mentality.

When developers build software removed from the customer impact, operations teams are left struggling to support applications without the time they need to develop a deep knowledge of any of them. If customers need support, it makes it difficult for operations to help them.

DevOps brings these two teams together and focuses on building a partnership between the two to reduce the likelihood of an outage and the recovery time in the event an incident occurs. By monitoring incidents as part of the software lifecycle and bringing it to the forefront, you can ensure that your customers are supported post-release.

5

OPERATE

IT has always been critical to the success of your organization. DevOps prioritizes IT service management (ITSM) – the process IT teams use to manage the end-to-end delivery of IT services to your customers. With the service provider model, operations can use platforms to automate more processes and set up more guardrails so that development teams can operate safely at scale.

Finding the right balance here requires your teams to come together and decide on a service contract between the two teams.

6

RESPOND TO CONTINUOUS FEEDBACK

Your goal is to support your customers' needs, so listening to their feedback is a requirement. Because DevOps is dependent on you releasing your code iteratively, you need a feedback loop that sends information to your teams so that they can prioritize what new features and functionality they should build into the next release.

The faster organizations can learn from their incremental delivery of software, the better they can prioritize upcoming work and decide to press forward in a successful direction or pivot to alternate strategies.

So by simply shifting the way that you think about the software development lifecycle, you can create a more cohesive process that prioritizes your customers first.

Build a strong DevOps toolset with Data Center

If you were to audit all the tools that your development teams are using, you'd probably find that they vary widely. In order to apply DevOps processes across your organization, you need the right tools in your arsenal. And, it's important to call out that not every tool is going to fit every single need. When it comes down to building a strong toolset that can support different phases of the software development lifecycle, the products you do pick should be adaptable to all of your development teams.

Many organizations have adopted SaaS products, such as **Atlassian cloud**, because they provide the added benefit of native automation and tighter integrations with other tools you may want to use as part of your DevOps strategy.

However, some organizations aren't ready or can't leverage SaaS products today. If this is your organization, that doesn't mean that you can't unlock the same capabilities, such as an IT service desk or automation, in self-managed products. It does mean, however, that the products you use need to have more advanced capabilities to support administration at scale, while providing the functionality your teams need to apply DevOps.

DevOps product requirements

- ☐ **Scalability and performance:** To successfully implement DevOps, you want to make collaboration as easy as possible. One way to do that is by having your teams standardize on the same product – removing the challenge of products that don't integrate together. That means that your products need to be capable of scaling, while also maintaining their performance – even when teams are distributed.
- ☐ **Security and compliance:** Maintaining security is of the utmost importance for any organization, so you need to make sure that your products have the security capabilities that allow you to remain secure even as you accelerate delivery.
- ☐ **Infrastructure flexibility:** For self-managed products, having infrastructure flexibility is key. You need to have the ability to streamline deployments when possible and find more innovative ways to automate different tasks.

Data Center is our self-managed offering that gives you the ability to maintain control of how you run and host your data to meet your organization's requirements while offering the capabilities and features you need to implement DevOps successfully.

Performance and scalability

To address the core scale, availability, and reliability needs of enterprises, we've built scaling capabilities directly into the platform. These capabilities make it easier to support teams at scale by optimizing the overall performance of the product and cutting down time spent administering the products. To learn more about some of the latest platform scaling enhancements we've made, read our **Community post**.

If high availability is a requirement, you can deploy Data Center in a clustered architecture. In a cluster, your user traffic is distributed through a load balancer to application nodes in your cluster, which allows you to scale your products more reliably.

Supporting your geo-distributed teams

Beyond the capabilities outlined above, Data Center also includes additional capabilities to keep your products performant and available at scale.

Because products are a key part of your DevOps strategy, you need to ensure that your teams, which are likely distributed across many locations, can access them anytime, anywhere. Data Center has support for **content delivery networks (CDN)**, which cache static resources locally so that all teams have the same performance, even if they aren't located close to your servers.

Security and compliance

Security is always top of mind for enterprises – this remains especially true as organizations like yours start to leverage more automation and speed up delivery to their customers. By choosing products like Data Center that have **security built directly** in out-of-the-box, you don't need to compromise agility.

To help you keep track of what's going on in your instance, you can use advanced auditing. This feature enables you to gain a security relevant record of the events that occur within your instance. And because most of you are already using some type of monitoring tool, you can leverage **file externalization** to integrate the two and gain even more insight into your data.

Infrastructure flexibility

By optimizing the infrastructure that your products are running on, you can take full advantage of the benefits of DevOps. Data Center has the flexible deployment options that you need to prioritize automation and maintain the scalability and reliability of your products.

First, you can decide whether you want to deploy Data Center in a clustered architecture, described above, or in a non-clustered (single node) **architecture**.

Then, you can choose to deploy your products on your own hardware or to leverage a cloud provider, such as AWS or Azure.

If you are looking to have a cloud like experience while still maintaining control of your data, leveraging a cloud provider is a great way to get the benefits of cloud computing on your terms. By deploying on **infrastructure as a service (IaaS)**, you can scale your products seamlessly – without having to worry about physical infrastructure. Plus, you'll get the added benefit of automation that's often built directly into the cloud platform.

Regardless of how you choose to deploy your products, we have **infrastructure recommendations** to help you optimize your infrastructure and support your teams' needs.

The **AWS Service Management Connector for Jira Service Management** enables your teams to provision, manage, and operate AWS resources natively via Atlassian's Jira Service Management. You can provide pre-approved, secured, and governed AWS resources to end-users through an AWS Service Catalog.

Modern capabilities for DevOps

Data Center makes it not only easier for you to manage your environment but also makes it easier for you to adopt DevOps practices with new, modern elements.

Keep your environments consistent

To keep your teams productive, maintaining consistent environments is crucial. However, this can be difficult because your development teams may make changes to their environments during the build process and you're tracking significantly more resources than you did in the past.

One of the ways that organizations have been able to overcome this is by using container images. When artifacts are built using older methods, code and infrastructure are managed independently, so code is dependent on the environment.

With containers, the underlying code along with libraries and environment configuration is bundled into an image that can be run on *any* computer that has a containerization platform. This is incredibly powerful because you've essentially solved the problem that occurs between environments not matching.

Most organizations use virtual machines (VMs) for their development and testing. For VMs to work, you need to make a complete copy of a guest operating system, whereas containers don't need it. These containers can also be packed more densely in a virtualized environment, which reduces space on your servers and consumes fewer resources than traditional VMs (huge bonus for your operations team!).

Allowing applications to be encapsulated in self-contained environments allows for quicker deployments, closer parity between development environments, and infinite scalability.

Our **Data Center Docker images** contain the static components you need to install Data Center. After defining your required configuration once, you can instantly deploy exact replicas of your environment from the command line at every stage of your deployment lifecycle, giving you the agility needed to keep valuable work moving forward and the flexibility to accommodate your organization's evolving development strategy over time.

If you need to make changes to access policies or add new compliance protocols, you can make these changes in one place and programmatically apply them across every instance. This eliminates the administrative overhead while maintaining a single source of truth for your development and production environments.

While using Docker images makes it easier for you to create consistent environments without the overhead incurred by traditional development methods, keeping track of all your images can still be challenging at scale. That's why many organizations using containers are also using an orchestration framework.

Orchestration frameworks, or systems, such as Kubernetes (K8s) act as an infrastructure for your container-based applications, allowing them to operate in a highly organized, secure environment.

With K8s, you also get the following benefits:

- Distribute user traffic to balance the load and maintain the stability of your container deployments.
- Mount the storage system of your choice, such as public cloud providers, or local storages, automatically.
- Automate your containers. Depending on your needs, you can create new containers, remove existing ones, or adopt all of their resources to a new container.
- Fit containers onto your nodes to make the best use of your resources.
- Restart, replace or kill containers that don't respond to your health checks.
- Store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Coming in 2021, you'll be able to use our official helm charts and Docker images to deploy your Data Center products and operate them in a Kubernetes environment – unlocking the benefits of the K8s platform and enabling you to deploy faster, scale easier, and operate your resources more efficiently than ever before.

Keep your environments updated

One of the most challenging parts of IT operations at scale is making sure that the products your teams are using stay updated and remain secure. Data Center offers **rolling upgrades and zero downtime upgrades (ZDU)** to make the process easier. With rolling upgrades and ZDU, you can upgrade your products without having to pull your system offline, which allows you to have a cloud-like upgrade experience that you control.

Rolling upgrades enable you to upgrade your instance to the latest bug fix of the version you're running, so you can easily pull in the latest security patches or critical bug fixes.

With ZDU, you can upgrade your instance to any version as long as it's on the same platform release. This means not only can you pull in the latest security patches, but you can also upgrade from one **long-term support release** to the next, as long as it's on the same platform release. This will enable you to take advantage of the latest features available in that release.

Automate tasks

Part of building strong DevOps practices is automating more of your tasks. One example of a task you can automate with Data Center is upgrades. You can automate and script rolling upgrades all through API calls.

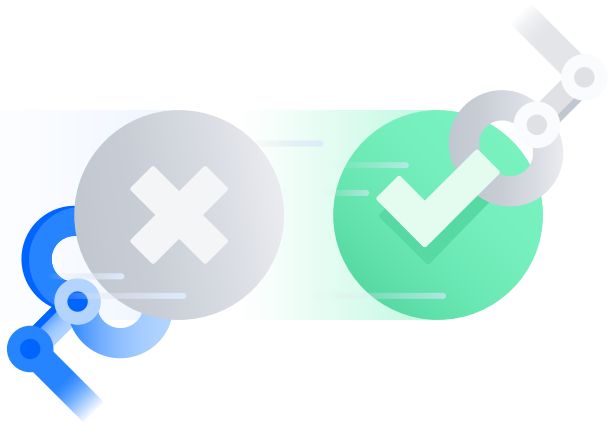
If you're deployed on IaaS, you can leverage infrastructure as code (IaC). IaC is a practice where infrastructure is provisioned and managed through code and software development techniques.

The API-driven model enables your developers to interact with infrastructure programmatically at scale without needing to manually set up and configure resources. This allows developers the ability to interface with infrastructure using code-based tools and treat infrastructure in a

manner similar to how they treat application code. Because they are defined by code, infrastructure and servers can quickly be deployed using standardized patterns, updated with the latest patches and versions, or duplicated in repeatable ways.

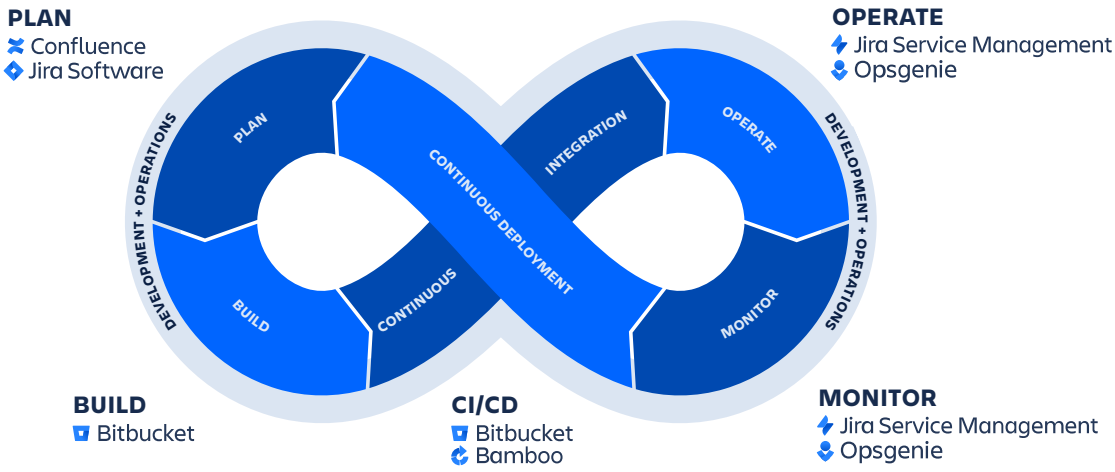
Using IaC can help you reduce disparity between environments that can exist when people manually configure their resources and when you script rolling upgrades alongside it, you can update your resources more efficiently and effectively.

To see rolling upgrades in action, check out our **deployment video library**.



How to use Data Center for DevOps

Now that we've looked at the foundational elements of Data Center, let's look at how you can use the products at each phase of the software development lifecycle.



1

PLAN

Confluence, Jira Software Data Center

You'll want to start with an idea or a goal for your release. With **Confluence Data Center**, you can create a project poster and use it to collaborate with your team so that everyone one is on board. This project poster in Confluence becomes your single source of truth – with links to roadmaps, progress tracking, and a dedicated space to track success metrics.

Once you've come up with the idea, you can use **Jira Software Data Center** to plan, track and release your software.

How Atlassian does it.

At Atlassian, we use Jira Software to plan and track our releases. This allows teams across the organization who aren't directly involved with the coding process to get status updates and make sure that releases go off without a hitch.

2

BUILD

Bitbucket Data Center

Bitbucket Data Center not only provides a central repository for developers to check in their code, but it also gives you the added functionality of pull requests. Pull requests allow all teams to collaborate on code by providing a location where developers can discuss the changes that are being committed. For example, you can post feedback in a pull request if there is an issue with the code, and a team member can tweak the code by pushing follow-up commits.

For teams using VS Code, you can integrate both Jira Software Data Center and Bitbucket Data Center and interact with them directly from your integrated development environment (IDE). Goodbye context switching.

To make it easier for developers, here are some additional capabilities you can use in Bitbucket Data Center:

- Track what has been reviewed and what needs attention
- Require multiple reviewers, successful builds, or passing security reports before allowing merge
- See related Jira work items at a glance, or create new ones based on the review

When you integrate Bitbucket Data Center with Jira Software Data Center, you can track your pull requests against your tickets for transparency. Developers can also create branches through Jira depending on their needs.

3

CI/CD

Bitbucket, Bamboo Data Center

Bitbucket's job doesn't stop at build. It becomes a critical piece of your CI/CD pipeline. Because you're already using Bitbucket, Code Insights gives you information, such as:

- Static analysis reports
- Security scan results
- Artifact links
- Unit tests
- Build status

> Integrate with Snyk to show security vulnerabilities in pull requests so developers can fix them before they push to production and avoid post-production debugging and security incidents.

This information helps you make informed decisions before you push your code into production.

With integrated CI/CD, you can then leverage **Bamboo Data Center** (coming soon) to create your build scripts. Much like Bamboo Server, Bamboo Data Center enables you to automate builds and unit tests, updates you on successful and failed builds, and provides you with the reporting tools you need for statistical analysis. If you're not using Bamboo, Bitbucket Data Center integrates with most leading CI tools. Either way, you have an end-to-end solution for all of your CI needs.

Integrated CI/CD in Bitbucket Data Center also enables you to get continuous feedback on your code in Bitbucket with a new Builds page and a new Builds tab on the Pull requests page.

4

MONITOR AND OPERATE

Jira Service Management Data Center, Opsgenie

While they are technically two phases of the software development lifecycle, some of the enhancements we've made in our products bring the two closer together.

Jira Service Management Data Center is our ITSM solution. You can use the service desk for both your internal and external teams. If your customers have any problems, they can submit tickets directly to your IT team when they need help. Your development teams can also use an internal version to request access to resources or applications.

And when it comes to monitoring outages, Opsgenie is your go-to app. Opsgenie is an incident management platform that provides

the capabilities to design alerts, manage on-call schedules, and keep your teams collaborating during the indecent resolution process.

Opsgenie is designed to be integrated with Jira Software and Jira Service Management. Now, you can link issues and service desk tickets to your incidents and alerts in Opsgenie so that your teams can be quickly notified
if there is an outage and resolve it quickly – without too much impact to your customers.

5

CONTINUOUS FEEDBACK
Confluence Data Center

One of the most important parts of DevOps is getting continuous feedback and applying those learnings to your next project. Confluence Data Center is a great resource to share information amongst your teams.

Confluence can be used as your knowledge base, where operations can share issues that may have occurred post-release. Additionally, your teams can use it for their retrospectives as part of the agile process.

Practice makes perfect

Building a strong DevOps framework takes time and patience, but with the right products, like Data Center, in place and teams championing these new practices and processes, you'll start to see the benefits of adopting DevOps.

Being able to bring your development and IT teams closer together is a big part of delivering value to your customers quicker, but you can't compromise security in the process.

02

Challenges security teams face

Challenges security teams face today

While DevOps has gained momentum and traction over the years, other teams have started to adopt similar principles – specifically security teams and operations. These two teams face some of the same challenges that DevOps aims to solve, but they're looking at them from a different perspective.

As an enterprise, you're faced with some of the following security challenges:

Increased security threats

According to Gartner, the average time to exploit a known vulnerability has decreased four-fold in recent years. Security risks continue to become more sophisticated and rampant across organizations. Furthermore, many companies that have experienced a security breach have indicated it could have been avoided with a patch or configuration change.

Common security threats can lead to:

- Substantial financial loss
- Intellectual property (IP) theft
- Disrupted user management

Insufficient resources

As systems become more complex, cybercriminals are only getting smarter. Unfortunately, this also means that dedicated IT and security resources and talent cannot keep up and are, therefore, unable to solve all of today's security challenges. Without assistance and shared ownership of security, organizations are left vulnerable to attack and internal error.

Differing priorities

Security and operations teams tend to have differing priorities. Security is focused on maintaining a safe and secure environment while operations is concerned with speed and performance. These conflicting goals make it difficult to work together, and the push for speed can encourage an optional mindset towards security, resulting in the release of vulnerability-ridden software and updates.

Traditionally, operations was responsible for setting up systems to achieve uptime and performance goals. Security was responsible for ensuring regulatory or compliance requirements, patching security fixes and combating security incidents as they arose. In this environment, security was a burden, slowing down operations and creating overhead. But in reality, security is a fundamental requirement of every IT system, just like uptime and performance.

Applying SecOps

To help organizations overcome these security challenges, a new framework was built: SecOps. SecOps is the collaboration between IT security (Sec) and operations (Ops) teams. This shared ownership approach prioritizes agility by breaking down silos to improve security and accelerate performance. Both teams work toward a common goal of delivering service that meets security and operational standards to improve business outcomes.

By enabling organizations to analyze security threats, adopt incident management strategies, and reduce breach response time, enterprises are seeing increased business security and optimized security controls. SecOps is comprised of practices, processes, and tools to ensure safe and secure application environments, meaning that organizations no longer have to compromise security for uptime and performance.

While the implementation of SecOps can be difficult, it is often a necessary step to maintain a competitive advantage and guarantee the continued security of your software.

THE BENEFITS OF AN OPTIMIZED SECOPS ENVIRONMENT:

- Fewer configuration errors are made
- Known vulnerabilities can be proactively managed
- Security and compliance policies are automatically checked and enforced
- Key security procedures are automated

Get teams to focus on security

For security to become a priority in your organization, everyone must be invested in your security strategy. A clear understanding of the challenges that you're facing allows your teams to take accountability and create processes to address these issues. This is where the behavioral shift begins.

One of the key concepts in SecOps is breaking down silos between security and operations. In SecOps, security and operations both share ownership (and responsibility) for secure operations processes. Teams that traditionally worked to achieve different goals now collaborate to reach common business objectives.

Build security practices

To successfully make the change, you should plan ahead and consider not only the strategy but the technical aspects as well.

There are two strategies that your organization can use when applying a DevOps framework:

- Reduce risk with automation and orchestrated processes
- Extend operations processes to support security

Reduce risk with automation and orchestrated processes

As cyber attacks proliferate, automation has become increasingly important for security teams. In an effort to manage incident response and streamline security operations, teams must be able to react quickly and efficiently. While many automation solutions are hailed as the gold standard for streamlining workflows and enabling teams to identify threats faster, oftentimes, these tools fail to effect change on the IT side.

By leveraging automation across every stage, SecOps teams can operationalize security across the organization. This can start with automated scanning for vulnerabilities, flaws, malware, and configuration mistakes and be extended to breach detection and response. This integration allows teams to reduce risk, gain insight into their environments, and maximize the efficiency of their security stacks. Automation gives both IT and security teams the opportunity to coordinate their efforts from the beginning of their workstreams all the way through remediation and closed-loop reporting. To make this happen, security must have access to the tools needed to identify issues and apply fixes or patches easily, while giving operations the visibility and control they need to test proposed security fixes.

Did you know? Jira Service Management allows agents to use a “low code, no code” approach. This functionality means that you can create automation rules within the platform to streamline workflows, delivering better support to end-users.

Get the most out of your security stack

One of the biggest risks that automation addresses is human error. Engineers and security teams alike can easily make mistakes performing repetitive tasks at hyperspeed. Unlike humans, software solutions are less likely to produce these same mistakes. Automation does not replace human teams, rather it frees them up to focus on business critical needs. While software can detect patterns, pinpoint bad actors and disseminate alerts, humans need to be involved to set up the automation rules and handle the unexpected.

The key benefit of automation is allowing SecOps teams to increase efficiency and prioritize business needs. So, instead of hiring more people to maintain security, automated and orchestrated processes can reduce risks by detecting and addressing threats faster. Having these workflows in place means that teams can more easily put policies in place that manage risk and minimize the likelihood of system breaches.

Gain visibility and see the bigger picture

Security teams are inundated with security alerts on a daily basis. In fact, many organizations are overwhelmed by the sheer number of alerts generated by their security teams. With automation, SecOps teams can identify patterns across their alerts and prioritize or “flag” them based on priority or concern. This strategic prioritization can help ensure that security teams can focus on more pertinent tasks such as addressing risks that will provide the highest ROI when mitigated. To take this a step further, organizations can also utilize automation solutions to gain insight into network operations. For example, automated processes,such as machine learning, can assist security engineers in understanding good user behavior, and thus making it easier to identify bad actors.

Barriers to automation

Legacy systems, which many security organizations still use today, rely on manual processes to function. While they are tried and reliable, legacy solutions are not built for modernization at scale. These systems require manual upkeep and feedback, slowing down day-to-day activities. Unlike automated or AI-driven software, legacy hardware stalls digital transformation and positions security teams at a disadvantage when moving to an operationalized model.

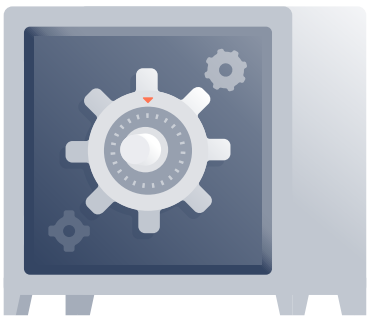
Your security teams use various technologies to manage security, and while these products may have automation available, security teams still need human intelligence to identify risks and exposure. But as the need to operationalize means key benefits for organizations, a shift is occurring.

Extend operations processes to support security

When approaching SecOps, it is necessary to understand both security and operations processes from end-to-end. Then, teams can begin the transition by gradually integrating security into existing operations practices. Building on current operations workflows, rather than creating an entirely new foundation, allows teams to easily introduce tools and technology so they can collaborate better and work more efficiently.

By using your existing IT operational processes as the basis for SecOps, it's easier to enhance security across the organization. Tasks performed by the operations team – including provisioning infrastructure, deploying applications and monitoring application instances – dictate how the entire IT organization operates. This access to IT resources makes it easier to extend security across multiple stages of your workflow.

Security teams, on the other hand, have a much narrower purview within the organization. Because they focus on things like scanning application code for vulnerabilities or monitoring environments for potential breaches, they don't have the same reach that operations teams do. By nature, their workstreams don't extend to areas such as application stage and provisioning. This means that, for SecOps, it's more practical to integrate security practices into the operations process than the other way around.



Process and products: Level up your SecOps implementation with Data Center

While getting buy-in from stakeholders is essential when it comes to realizing a cultural shift like the move to DevOps, the reality is that this change is not possible without the supporting technology. For DevOps, automation, infrastructure-as-code (IaaC), and the availability of resources are the enabling technologies that give both teams access to the speed and precision needed to “do DevOps.”

In the same fashion, once security and operations teams have decided they are ready to embrace SecOps, they must identify or build tools that enable them to work together and drive change.

Finding the right software to enhance SecOps can be challenging. You need to not only think critically about your current tech stack but also have alignment on your SecOps goals. This means evaluating your infrastructure and understanding how a solution will meet the specific needs of your environment. The right software will have robust capabilities and that meet your SecOps needs.

At Atlassian, security is built into the **core foundation of our products**. That’s why, when we think about SecOps, we consider software development, configuration, compliance, security incidents, and threat assessment. Our tools adapt to evolving SecOps teams’ needs and work together in order to foster a collaborative and holistic approach to security. These tools, alongside the Atlassian Marketplace, which hosts thousands of applications for IT/Ops teams, bring security protocols to the forefront and drive successful end-to-end SecOps practices.

Data Center offers features and capabilities to support your demands. As you work towards creating a well-defined SecOps strategy, Data Center empowers your teams with mission-critical applications to better safeguard your people and data as you operationalize security.

Automate user management

As an admin, you want to make sure that your products are always available for your teams while ensuring that it's done in a secure manner. With Data Center's advanced user management features, you can automate your **authentication and authorization protocols** to manage your users more easily at scale.

Single sign-on (SSO) is a great solution for managing account access and creates a seamless experience for end-users. More importantly, SSO mitigates security risks caused by the growing number of applications and logins as you scale. Atlassian's support for SSO enables features such as just-in-time provisioning, centralized management of authentication policies, and automatic lockout when a user is deactivated from your SSO provider.

Additionally, automated user provisioning allows for a direct connection between your identity provider and your Atlassian products. Data Center's advanced user management capabilities allow you to oversee user-related activities and easily achieve simple and secure authorization and authentication. This means that you have the power to manage user identities via a centralized view and can provision and remove users on-demand. The ability to de-provision users reduces the risk of security incidents by removing access for those that leave your organization. Gone are the days of manually creating and deactivating user accounts each time someone joins or leaves the company. These advanced capabilities give you the control and visibility you need, saving you time and ensuring your products' security.

Prioritize incident management

Even though you've safely gotten your teams access to their products, you still need to ensure that they are using their products in accordance with your security practices. With Data Center's advanced auditing features, you have access to event logs that track specific activity within your instance in the event of a security incident. Audit coverage, including security events, global permissions, and local configurations, allows you to track and monitor security threats in real time. This increases the speed of security breach investigations and gives teams the insight needed to act quickly and maintain a safe and secure environment.

By integrating your monitoring tools for long-term storage through file externalization, you can ensure a secure and tamperproof record which can assist in detecting security anomaly patterns. Additionally, features within your monitoring tools allow you to track trends and glean even more insight into your products. With the enhanced UI and improved tracking, you can easily visualize information and understand how products are being used across the organization.

When your monitoring tool flags an incident, Opsgenie is Atlassian's modernized incident management platform for operating always-on services, empowering dev, operations, and security teams to plan for service disruptions and maintain control during incidents. By integrating with over 200 monitoring, ticketing and collaboration tools, Opsgenie centralizes alerts, notifies the right people at the right time, and enables them to collaborate and take action quickly. Throughout the entire incident lifecycle, Opsgenie tracks all activity and provides actionable insights to improve productivity and drive continuous operational efficiencies.

Additionally, Opsgenie's incident templates allow teams to design incident response protocols and define workflows based on incident priority. This allows stakeholders across the organization to resolve problems quickly and collaborate effectively. These features are critical for maintaining transparency and creating a source of truth for incidents. Teams can rest assured knowing that there is a comprehensive record of incidents and a place to get up to speed and check ongoing statuses.

Increase visibility between security and software

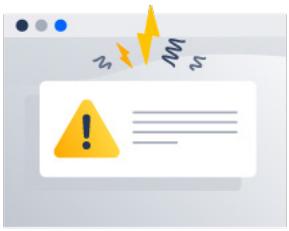
Having a digital record of everything that happens in your instance helps you make more informed decisions on how to administer your teams. It also helps when your leadership team looks to you for data that validates your organization's security posture. Data Center's **Data Pipeline** – coming soon - unlocks actionable data and insight for stakeholders that they can easily access to make data-driven decisions at the organization-level. Product-specific data can easily be exported to your BI tool of choice (e.g. Excel or Tableau), and integrations with data visualization tools translate your data into impactful and easy to consume data visualizations.

Additionally, Data Center's rate limiting feature allows you to monitor and control the rate of incoming and outgoing traffic, making your instance more secure. With self-protection capabilities, you have added visibility into who is being rate limited, how often requests are being limited, and when they were last limited. Custom configurations give you the ability to adjust user limits, protecting your instance against certain users and giving advanced permissions to those who require it. These insights help to prevent performance degradation and safeguard your instance from potential threats.

How to use Data Center for SecOps

Maintaining the security of your organization and of your customers is a must for SecOps. Here's an example of how Atlassian tools foster a truly collaborative, end-to-end experience for your SecOps needs.

Prepare and prevent incidents



Use Confluence Data Center as a single source of truth and house pertinent security documents, such as process guides, runbooks for troubleshooting, and retros. This allows for better collaboration and ensures that information is usable, shareable, and available to team members that need it.

It's also important to set up guardrails in case an incident occurs. Opsgenie enables teams to stay ahead of security incidents and better coordinate. With centralized alerts and incident management services, critical incidents can be prioritized and responded to and investigated in a timely manner.

Once alerts are in place, your security team can use Jira Service Management to streamline support with automation and set up workflows to route issues to the appropriate development team. Understand where you are in the workflow, tailor metrics and priorities that matter to your team, and better respond to security threats.

Resolve

If a security incident does occur, Jira Software Data Center can be used cross-functionally to manage project work and link IT tickets to get to the root cause of problems before they escalate. Jira makes it easy to configure automation rules for optimized incident management tracking.

Your security team can also improve security response and insight with Atlassian Marketplace. Access to over a thousand security applications means that integration across products is seamless and necessary protocols, such as incident monitoring and post-incident review (PIR), are second nature.

Iterate and learn

All successful processes stress iterating and learning. Return back to Confluence to create a PIR and response plan. Confluence provides an environment for cross-functional teams to collaborate and standardize processes via templates. Teams can even export Opsgenie PIRs right into Confluence pages for efficient reporting and added visibility.

Find the balance

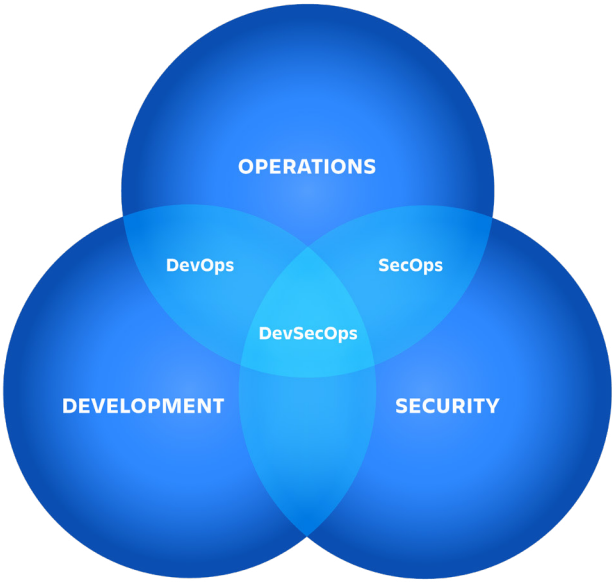
Different organizations are at different stages of developing their security presence. When deciding whether you are ready to operationalize your security efforts, you should think about the shifts in culture and technology that will get you there. SecOps is not a “trade-off”, rather it is about enhancing processes that best serve your business and customers. Speed and security are not mutually exclusive, and you can effectively integrate security into operations processes by working together as one.

03

DevSecOps: A natural progression

DevSecOps: A natural progression

By this point, you may be wondering, can you really have DevOps without SecOps? While the frameworks were designed to work independently of each other, many organizations have combined the two into a hybrid framework: DevSecOps.



With DevSecOps, security is embedded into all facets of DevOps including continuous integration, delivery and deployment. The principles that brought DevOps to popularity are now extended to include security. In fact, DevSecOps is a natural progression following a successful DevOps approach, focusing on collaboration and, of course, security. With a DevSecOps framework in place, organizations approach security as a proactive component of development as opposed to an afterthought.

Now that development, IT operations, and security are cross-functional and working towards a common goal, business practices and customer satisfaction can be optimized. By adopting security as code and integrating stakeholders into the development pipeline, you enhance internal workflows as well as the finished product. DevSecOps combines existing workflows with automated tools to drive value across the organization. When security verification becomes an active and integral part of the development process, agility and transformation are even more tangible.

Defining cultural expectations

Just like DevOps and SecOps, there are technical aspects to build this framework, but cultural transformation is where it begins. DevSecOps requires organizations to rethink people, process and products. Organizational change will support the progression needed to turn conventionally siloed departments into cohesive, highly functioning teams.

Take this example – within enterprise organizations, your security team's goal is typically to keep environments stable and minimize risk, while your developer's mindset is more geared toward speed. But let's consider security vulnerabilities:

When we talk about security vulnerabilities, your security is likely to see speed as a good thing because the faster they can patch and address these vulnerabilities, the faster they can rotate credentials and change out passwords in their environment. This spin on speed and security can be fostered through automation and DevOps practices on the development side.

This shift in thinking means that your developers can get to the point where they can push a new code change (with the latest vulnerabilities patched) into production having never seen a password and automatically get a new password when they need it.

At first blush, these teams appear to be at odds. But when you take a closer look into why stability and speed are important, you can begin to blur the lines and see that there's much more alignment than you initially thought. From a cultural standpoint, by focusing on this kind of small win, you can effectively start to build the trust that's needed across these teams.

While organizations with a well-developed DevOps environment have a leg up in the transformation to DevSecOps, there is still work to do. As development and operations cultures are strengthened to embrace DevOps, there is an additional cultural transformation that needs to occur in order to fully transition to DevSecOps.

Here are a few things to consider:

- Position DevSecOps as a next step
- Move from gated processes to shared responsibility
- Communicate about security across the organization

Clear expectations are vital to successful DevSecOps outcomes. It takes time to build trust and cohesion, so it is important to take all facets of this new framework into account without cutting corners. When incorporating security into DevOps, team members should consider the following:

Transparent communication

Team members across dev, IT operations, and security need to openly communicate to address security challenges related to infrastructure and design decisions.

Security team responsibilities

Security team members are actively involved in documenting and clarifying security expectations to the greater organization via security testing, equipping developers with the knowledge to write secure code.

Shared responsibility

With a collaborative approach, every team member is held accountable for security posture.

Test-driven security

Integrate security policies and testing into CI/CD testing prior to application deployment.

Unit-testable security requirements

Adherence to security standards becomes consistent and repeatable via unit-testable modules defined by the security team.

Iteration

Perform iterative tests across multiple products to ensure secure delivery.

Early failure detection

Security vulnerabilities and missed controls are detected early and quickly. This allows developers proactively fix issues as opposed to slowing down the development cycle with retroactive corrections.

Bring security to the forefront with a shift left approach

Traditional software development cycles typically place security testing to the end of the process. Code reviews or penetration tests are performed to identify security flaws late in the cycle as applications are nearing deployment. This approach is troublesome because it leaves room for vulnerability-ridden applications to get to the end of the pipeline before ever being noticed. Development teams are heavily relied on to create secure code while security is a retroactive function, added on the backend. This approach causes tension and disconnect between not only security and operations teams but development teams as well.

Security and development team members need to be on the same page in order to create secure code. To do this, dev teams need to be trained to ensure that testing practices are in place and vulnerabilities are addressed early on. Then, any flaws that are identified can be fixed and teams can move on to application deployment.

Move forward with a left shift

By inserting security into the early stages of your operational and development processes, you can accelerate delivery timelines and foster an environment of cross-collaboration. With this approach, security doesn't slow delivery down, and you can tighten up the feedback loop when an issue arises. Security teams can assist in creating secure code and, in turn, facilitate successful, continuous delivery.

Consider this. Rather than rewarding DevOps for getting code out fast and praising security teams for secure controls, consider celebrating DevOps for releasing secure code and acknowledging security for moving faster. This shift in mindset will empower both teams to see tasks from the other's perspective and align on goals.

When looking at the development cycle as a whole, shifting security left enables healthy business practices and prioritizes customer value. Rather than security teams catching issues, opening tickets, and waiting for developers or operations to fix things, they become empowered to address problems from the start. This not only saves time and people hours but also ensures that code can be released both continuously and securely. All teams involved need to share responsibility and take an active role in learning how other teams function, so they can be more tightly integrated into the pipeline.

A "Shift Left" Scenario Example

Let's walk through a standard enterprise software deployment. To initiate the process, teams will request servers based on organizational requirements and estimated user tier. The servers are then built (either physical or virtual) from existing templates fit to meet their needs (e.g. **Atlassian CloudFormation templates**). The application is then created and installed. Finally, security testing is performed prior to production. At this point in the process, the security team is introduced to identify any issues and, if found, they could delay the application from moving forward.

Now the question becomes, do they delay the project to meet security requirements, or instead, do you release the application, meet your deadline, and have the security team fix the problems later?

Instead, with shift-left, security and operations are active and engaged in the deployment cycle and co-create templates that are version-controlled and meet security requirements. Because there is shared accountability from the start, security teams are able to provide guidance during the build, and operations can offer insight into how to appropriately implement security controls. This accelerates the overall timeline because the legwork is done and templates are standardized prior to any strict delivery deadlines. Now, when there is a server requisition, the organization is already starting with a secure configuration. Security and operations work together during the install, and security flaws can be promptly identified and remediated.

How to start shifting left

While the benefits of a shift-left may seem obvious, it's easier said than done. When it comes to implementation, there are several key steps to get you on a path to success.

1

DEFINE YOUR SHIFT-LEFT SECURITY STRATEGY

Any foolproof plan begins with a thoughtful strategy. Goals and objectives must be clearly defined at the onset to ensure alignment across development, operations, and security teams. This means that your organization must define what a shift-left means to them. Key elements of this strategy include ownership and responsibility, milestones, and metrics. Note, this strategy document will evolve over time – teration is necessary for a strong foundation.

2

UNDERSTAND WHERE AND HOW SOFTWARE IS CREATED IN YOUR ORGANIZATION

Where and how does your organization create software? Depending on the size of your organization, this can be an easy question to answer or an extremely difficult one. This step allows security teams to understand where and how they can move security closer to development.

Completing this step gives security teams visibility across the organization and helps codify the flow of software. Start at the macro level and then to individual lines of business. Software development processes and tools likely differ across business units making this insight even more important. Key items to identify in this phase include who is developing code (people), how code reaches production (process), and which systems enable the process (technology) – also known as the CI/CD toolchain.

3

IDENTIFY AND IMPLEMENT SECURITY PROTOCOLS AT EVERY STAGE OF DEVELOPMENT

While quality assurance is embedded into the software development lifecycle, this process has not historically included security. For a successful DevSecOps program, this must change and the previous steps will act as a foundation. Security testing, feedback, and guidance should be provided at every step of the software development process, paying special attention

to security risks and vulnerabilities. By leveraging security protocols, development teams are armed with the tools they need to routinely create secure code.

4

CONTINUOUSLY TRAIN DEVELOPMENT TEAMS IN SECURE CODING

Developers know how to code, but do they know how to do it securely? Part of your journey to shifting security left is to ensure that those who do the majority of your coding create secure code from the start. This is difficult to do if you have no benchmark to measure your skills today and no plan to continuously improve them. Development teams must be trained, and security teams must be open and willing to impart knowledge around threat landscapes, secure coding etc. This process will not happen overnight, but, in time, your organization will see the benefits.

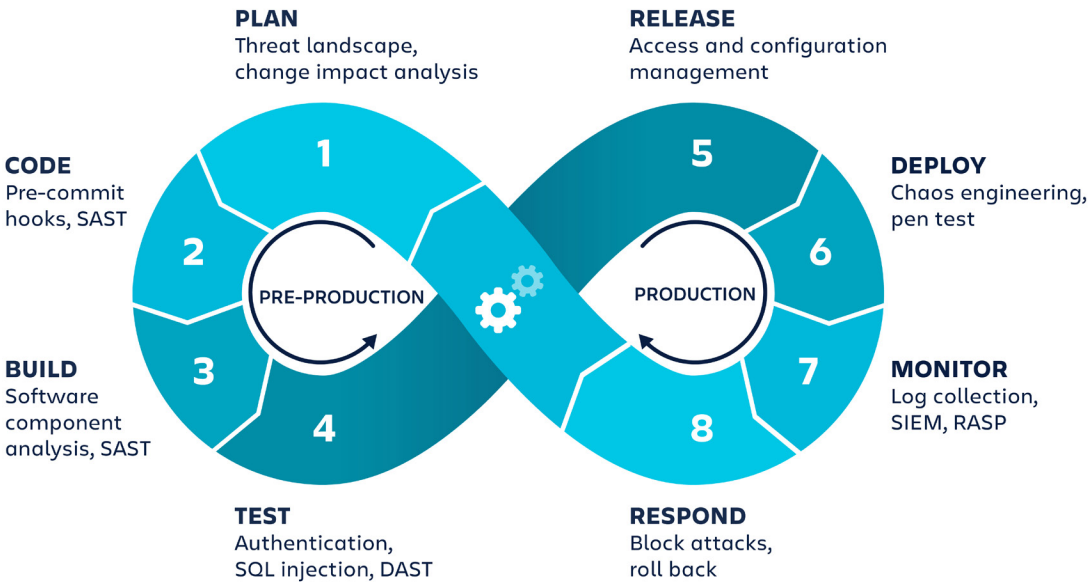
QUICK TIPS TO HELP TRAIN DEVELOPMENT TEAMS:

- Use the same tools
- Align incentives
- Provide access and shared ownership

Successful execution in a DevSecOps era

The need for safe and secure software is critical to the health of your organization. DevSecOps principles foster collaboration and allow for security measures to be pulled forward in the development process. Without DevSecOps principles in place, security is tacked on to the end of the development pipeline, which increases costs and delays releases every time a vulnerability is found. With a well-defined DevSecOps approach, security controls are baked into product development from the very beginning. This means a streamlined product development process with fewer security-related delays.

With DevSecOps, security is applied to each phase of the DevOps pipeline: plan, code, build, test, release, and deploy. This means continuous feedback at every stage of the process rather than ad-hoc tests or scheduled deployments. And more importantly, integrated continuous security approaches can scale as your business matures. To implement these principles, you need tools that will accelerate transformation and improve the pace of development all while maintaining security.



1

PLAN

While the plan phase is the least automated phase of DevSecOps, it is critical to a successful release. This stage prioritizes security by embracing collaboration, discussion, review, and strategy of security analysis. During this phase, teams are defining user stories and outlining how, where and when testing will occur. This phase helps to address security vulnerabilities and improves the security knowledge of everyone on the team.

2

CODE

In the code phase, DevSecOps tools ensure that developers write secure code. Security practices in this phase detect violations in coding best practices and identify security vulnerabilities in code. Practices to implement at this stage of the delivery pipeline include static analysis security testing (SAST), code reviews, and pre-commit hooks.

3

BUILD

The build phase begins once developers commit code to the source repository. At this stage, development teams are empowered to remediate critical and high risk security issues as they arise. By performing advanced automated testing of the application (unit tests, risk-based security tests, SAST, etc.), teams can identify when code doesn't compile, SAST failures, vulnerabilities, etc.

4

TEST

The test phase begins after the artifact is built and deployed in a testing or staging environment. This stage uses dynamic application security testing (DAST) tools, which analyze applications from the outside and scan for things such as user authentication, authorization, SQL injection, and API-related endpoints. Additionally, security-focused DAST analyzes an application against a list of known high-severity issues.

Pro tip: Be sure to review the test automation tools and resources available on the **Atlassian Marketplace.**

5

RELEASE

Just before releasing the application, utilize security analysis tools to perform thorough penetration tests and vulnerability scans. This will ensure that you've addressed any security issues before rolling it out on production.

6

DEPLOY

Because your code has been delivered to your customers, it doesn't mean that it will always be secure. Everyday, new vulnerabilities pop-up. Continue to test your code against these new vulnerabilities so that you can make any necessary changes.

DEVSECOPS BEST PRACTICES

While shifting team culture from DevOps to DevSecOps might seem intuitive, it can slow the development process at first.

Here are several best practices to consider when implementing a DevSecOps framework so you ensure that your team is prioritizing security and agility:

- Provide security training
- Equip your teams with the appropriate tools
- Address the entire application delivery pipeline

Building a successful DevSecOps framework with cohesive teams is an iterative process. Much like the methodology itself, cultural development and integration requires continuous adjustments to achieve organizational goals and objectives. Understanding the long term implications and establishing open channels of communication and cultural expectations are all key steps toward a successful DevSecOps program and broader digital transformation.

Curious to see the products in action?

Download free trial licenses of our Data Center products to see how you can use them as you implement these frameworks in your organization.

Ready to get started?

Download any of our Data Center products today to try it for free!

- Bitbucket Data Center
- Crowd Data Center
- Confluence Data Center
- Jira Service Mangagement
- Jira Software

ATLASSIAN