The Salesforce DevSecOps Guide

Build, test, monitor, and govern your Agentforce and Customer 360 securely on the Salesforce Platform.

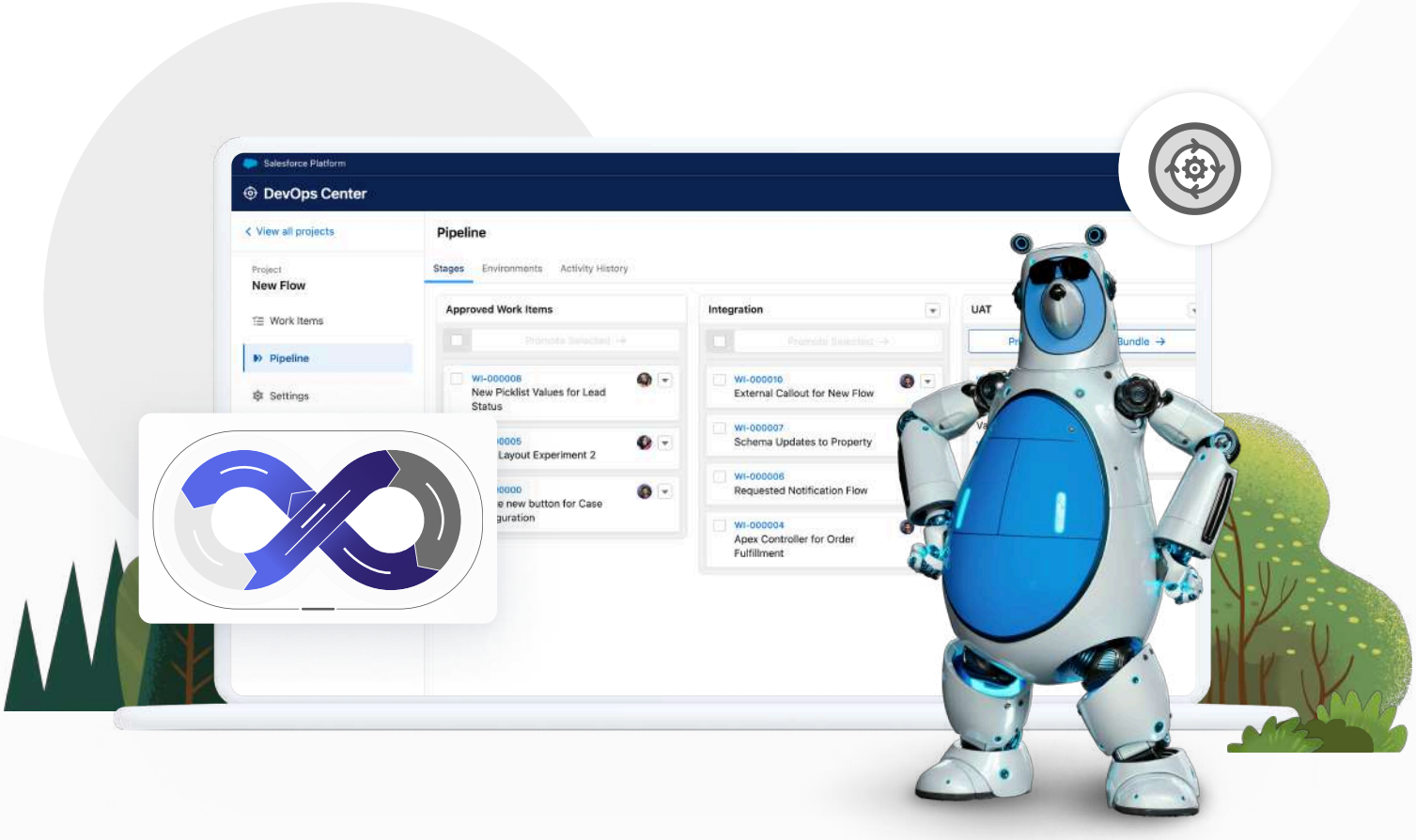


Table of Contents

Section 1

Why You Need a DevSecOps Strategy

We're in the third wave of AI where AI agents aren't just helping us work smarter; they're working autonomously, making decisions that actively shape organizations. Businesses are now racing to build and deploy autonomous agents and AI-powered experiences at scale – not just to enhance workflows, but to transform how they engage customers and empower employees. In this rush, traditional development lifecycles are struggling to keep pace. And the cracks are starting to show: 29% of projects are now missing deadlines, up from 26% last year.

This push toward agentic AI adoption is increasing complexity for many organizations. And without a strong security foundation, the rush to deploy automations, AI applications, and agents can put sensitive data and customer trust at risk. The challenge? The more AI scales, the harder it becomes to keep security, transparency, and governance in check. Staying ahead means rethinking our approach – one that keeps innovation moving at full speed without cutting corners on security.

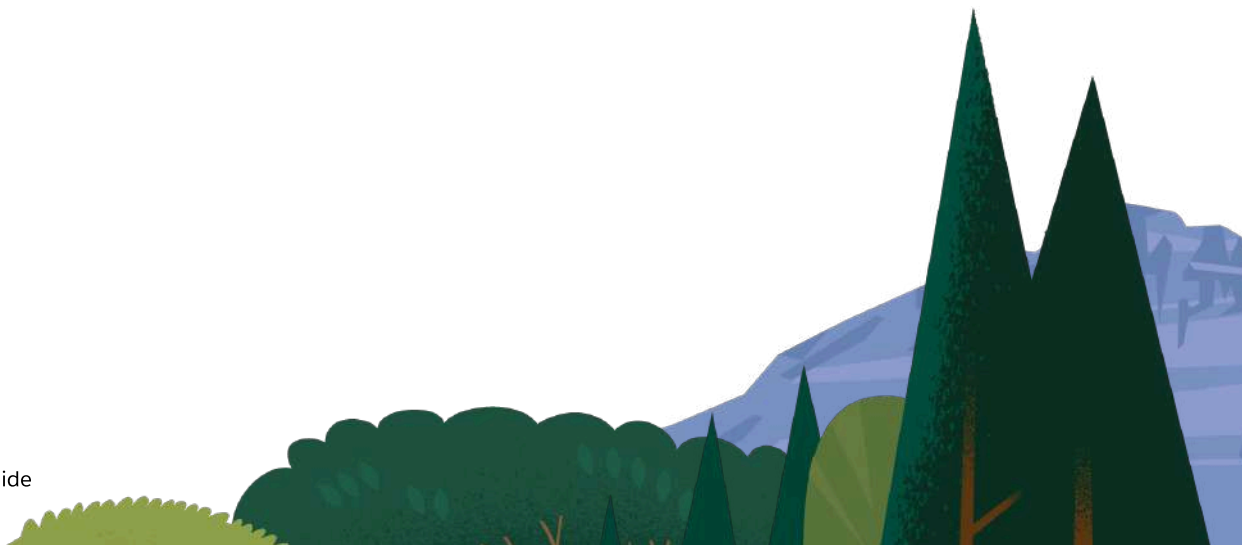
AI generates more data than traditional approaches to security can handle

By 2027, the world will store nearly 300 zettabytes of data – so much that it's hard to even wrap our heads around. This is more than just a shift in technology; it's a major responsibility to manage and protect that data. Businesses will be key in ensuring it's stored securely, used effectively, and drives innovation.

As organizations increasingly lean on data to power AI-driven experiences and stay competitive, the need to protect that data has never been greater. And the concern is growing – 75% of security leaders believe AI-driven cyber threats will soon outpace traditional defenses, and 29% say their security practices must evolve as AI adoption grows.*

With the increasing use of AI – and the explosion of data that fuels it – this challenge is only growing. Ultimately, AI needs to be built on a foundation of trust, because without it, data stops being an asset and starts becoming a liability.

* Salesforce State of IT: Security Report



Lack of transparency will undermine AI trust

Businesses see data as a strategic asset, but customers remain skeptical. In fact, 65% of customers believe companies are reckless with their data. At the same time, 79% say they would trust organizations more if data usage was clearer.

The takeaway? For IT leaders, this means security and transparency must be embedded into solutions their teams are building – including automations, AI apps, and agents.

AI implementation cannot outpace security readiness

IT teams are under immense pressure to deploy AI innovation across the organization – and to do it fast. But behind the scenes, many teams are failing to keep up. In fact, 88% of IT teams are struggling to keep up with the flood of AI-related demands.

What's slowing these teams down? Well, there are several key challenges they are facing when it comes to AI adoption:

- **Data fragmentation and compliance issues complicate AI development**
- **Lack of a clear AI security strategy creates vulnerabilities**
- **Governance gaps leave businesses without structured processes for responsible AI deployment**
- **Rigid platforms prevent AI scalability**
- **A widening skills gap makes it harder for teams to implement AI securely and efficiently**
- **Increased costs of programming and tools put additional strain on resources**

And the stakes are only getting higher. With IT services spending expected to grow by 9.4% this year, hitting $1.73 trillion – much of that driven by AI-related projects – the pressure to deliver quickly is real. But in the race to deploy AI solutions, cutting corners on security isn't just risky, it can be costly.

That raises a critical question: as innovation speeds up, what's being prioritized – and what's being sacrificed?

To move fast without compromising trust, organizations need a better way forward. **DevSecOps** offers that path. By embedding security into every phase of the development, DevSecOps helps IT teams build AI solutions that are not only fast and scalable, but also secure and resilient from the start. It's how innovation stays aligned with integrity – and how organizations can keep pace without putting themselves at risk.

Section 2

Avoiding the Security Checkbox Paradox with DevSecOps

Speed and agility are the heartbeat of modern software development. DevOps changed the game for IT teams by tearing down silos between development (“Dev”) and operations (“Ops”) teams, making it easier to roll out updates, fixes, and new features at the pace customers and employees expect.

But while DevOps streamlined the process with automation and continuous integration/continuous delivery (CI/CD), security often got left behind – treated as a final checkbox rather than a core priority. This led to more vulnerabilities, compliance headaches, and costly fixes down the line.

That’s where DevSecOps changes the equation. By bringing development, security, and operations together from day zero, it builds a safety net into every stage of the software development lifecycle (SDLC) – a practice called "shifting left." Instead of scrambling to patch security gaps late in the game, teams catch and fix issues early, when they’re faster, easier, and cheaper to resolve.

85% **of IT organizations follow a DevSecOps model.***

Organizations with DevSecOps are more confident as they build and deploy AI agents.

* Salesforce State of IT: Security Report

With DevSecOps, security isn’t a roadblock – it’s part of the workflow. Teams proactively test, monitor, and secure applications in real time, including AI-driven solutions like Agentforce. Automated security checks, traceable workflows, and continuous monitoring help keep threats in check, while features like isolated testing sandboxes, data masking, automated anonymization, and source tracking add extra layers of protection. And when it comes to compliance, automation makes auditing seamless, creating a clear, traceable record of every action.

The value of DevSecOps

DevSecOps isn't just an upgrade to DevOps – it's a smarter way to build and scale automations, AI apps, and agents securely. With AI relying on massive amounts of real-time data, trust and compliance are non-negotiable. DevSecOps ensures that both customer and company data stay protected while giving businesses the freedom to scale automations, AI apps, and agents with confidence.

At its core, DevSecOps is about moving fast without breaking trust. By baking security into the development process from the start (shifting left), teams can innovate at speed without having to put data integrity on the line.

Organizations that follow a DevSecOps model are:

20%
more likely to
implement AI securely

31%
more likely to
deploy AI with the right permissions and protocols

25%
more likely to
deploy AI on time to meet business goals

20%
more likely to
be compliant with regulatory requirements

* Salesforce State of IT: Security Report

Data breaches are becoming more frequent – and more expensive. Cybersecurity incidents don't just put sensitive information at risk; they're now one of the leading causes of IT downtime, disrupting operations and driving significant costs for businesses. With AI-powered systems increasingly at the core of how organizations operate, the stakes for getting security right have never been higher.

Teams can minimize these risks by taking a comprehensive, proactive approach to security and making sure it's built into every stage of design, development, and deployment. That means creating secure environments where new features, integrations, and changes can be safely tested before going live – including using masked data in pre-production to protect sensitive information. When these foundational security practices are combined with automated vulnerability scanning, teams are better equipped to catch issues early and address them consistently – ensuring smooth rollouts without introducing vulnerabilities or costly disruptions.

But it's important to remember that security isn't just about avoiding breaches – it's also about maintaining compliance. As data protection and privacy standards evolve, organizations are spending more time on audits and regulatory readiness.

Teams can implement common controls that support compliance with multiple region- and industry-specific regulations – including the International Organization for Standardization (ISO) and the National Institute for Standards and Technology (NIST), along with laws like the Sarbanes-Oxley Act (SOX), Health Insurance Portability and Accountability Act (HIPAA), and EU Artificial Intelligence Act (European AI Act). DevSecOps makes it possible to build with security and compliance in mind from day zero.

ISO Compliance

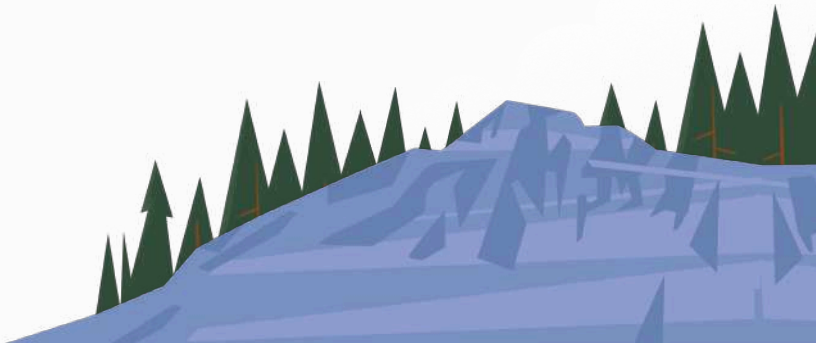
The International Organization for Standardization (ISO) is a Geneva-based NGO that has published some of the best-known standards for global industry best practices. ISO has released more than 22,000 standards, including ISO 27001, which outlines a very specific set of strategies and checklists for creating strong security measures across an organization.

HIPAA Compliance

The Health Insurance Portability and Accountability Act (HIPPA) was passed by the Department of Health and Human Services Office for Civil Rights in 1996 to protect citizens' individually identifiable health information. These standards ensure that healthcare organizations and their business associates know how to handle patients' sensitive data, which is formally defined as protected health information (PHI).

But it's not just about knowing how to handle sensitive data – HIPAA also has specific requirements for protecting and monitoring access to PHI. Integrating these requirements into DevSecOps practices helps teams build compliant systems from the start, without sacrificing speed or agility.

Learn more



Deliver faster and safer with DevSecOps best practices

Security isn't a trade-off for speed – it's a catalyst. Secure-by-design accelerates development by reducing rework and delays. As teams roll out innovations like Agentforce and Data Cloud or push updates to existing applications, the pressure to ship fast can't come at the expense of trust. That's where DevSecOps comes in.

DevSecOps embeds security into every stage of development, enabling teams to release quickly and safely. Here's how to build strong DevSecOps habits into your workflows:

Shift Security Left

Security shouldn't wait until the end of the development cycle. Instead, "shift left" by embedding security practices early – during design, development, and testing. This includes integrating data masking and anonymization, threat modeling, secure design principles, and static code analysis (SAST) right from the start, when vulnerabilities are cheapest and easiest to fix.

Automate Security Testing in the CI/CD Pipeline

Manual reviews can't keep up with the pace of development – especially at the rate AI innovations are being demanded. Automate testing for vulnerabilities, misconfigurations, and policy violations at every stage of your CI/CD pipeline using tools like SAST, DAST, and Software Composition Analysis (SCA). This helps catch issues immediately – without slowing teams down.

Build in Compliance from the Beginning

Classify data by sensitivity level and relevant regulations early in design and planning, then embed compliance as code and validate it continuously. Automating these checks ensures every build meets standards like GDPR or HIPAA, making it easier to prove security readiness and reduce audit overhead.

Perform Risk Assessments

Start by identifying and prioritizing the highest risks, then implement role-based access controls and limit permissions to only what users and services actually need. Least privilege minimizes potential blast radius in the event of a compromise and aligns with zero trust principles, which are critical in AI-driven environments handling sensitive data.

Maintain Visibility Across the Pipeline

Security is a team sport. Use tools like DevOps Center or version-controlled pipelines that offer traceability, dashboards, and logs – so security, development, and operations can all see what's changing and when. Transparency reduces errors and supports faster remediation.

Continuously Monitor and Remediate

Security doesn't stop at deployment. Continuously monitor automation, apps, agents, and infrastructure for vulnerabilities, anomalies, and compliance drift using runtime security tools. Pair this with alerting and automated remediation workflows to reduce time-to-fix.

Foster a Security-First Culture

Equip developers with secure coding knowledge through training, feedback loops, and tools embedded in their workflow. Create security champions in each team to promote shared responsibility and faster issue resolution.

Section 3

Putting DevSecOps into Practice

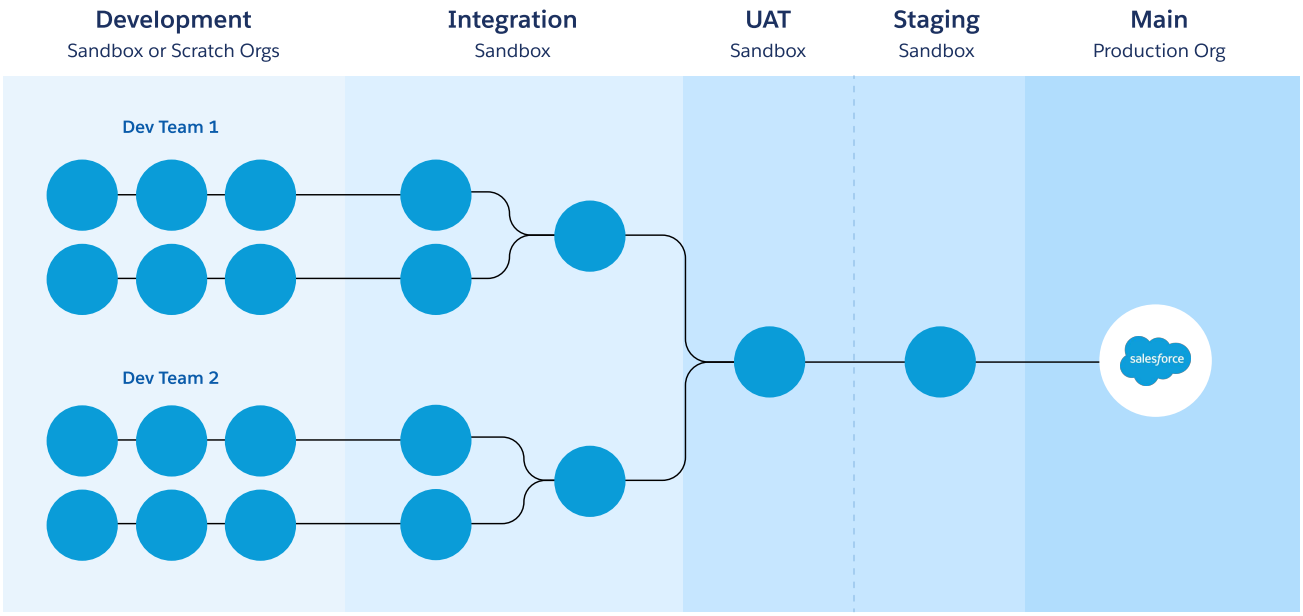
A DevSecOps approach brings together business users, developers, and admins in a secure, automated environment. With open tooling and simplified workflows, teams can plan, build, test, release, and monitor solutions faster – without sacrificing security or quality.

A well-structured branching strategy is key to managing changes securely and efficiently. Think of your branching strategy as a roadmap that makes sure that every change is tested and validated before it hits production. And each branch represents a different stage of the application lifecycle – here's how it typically flows:

- **Development branch:** Teams build and refine new features in isolated branches
- **Integration branch:** Changes are merged and automatically tested to catch issues early
- **UAT branch:** Once integration tests pass, updates move to user acceptance testing (UAT) to ensure they meet business needs
- **Staging branch:** Successful changes are deployed in a near-production environment for final validation – including a security audit or risk assessment to prioritize high-risk issues before going live
- **Main branch:** After passing all checks, updates are securely merged into production

By following this structured approach, teams can spot security risks early, stay compliant, and roll out updates smoothly – so security and innovation can work together, and not against each other.

Development Across the ALM Process



Development across the agent and application lifecycle management (ALM) process

Let's look at an example of how a team would execute development throughout the ALM process.

DeeDee
Declarative Developer

- Manages applications using declarative tools.
- Pushes changes to version control systems and performs promotions through a secure graphical interface.

DeeDee is comfortable using user-friendly interfaces to interact with development processes. With a modern DevSecOps platform, she can commit changes with a few clicks, while the platform automatically handles source control synchronization behind the scenes. She follows built-in security prompts and validation rules, ensuring compliance before any commit.

Pedro
Programmatic Developer

- Manages applications using programmatic tools.
- Pushes changes, performs reviews, merges, and deploys using code editors, CLI, or version control systems. Changes are reflected in the graphical interface.

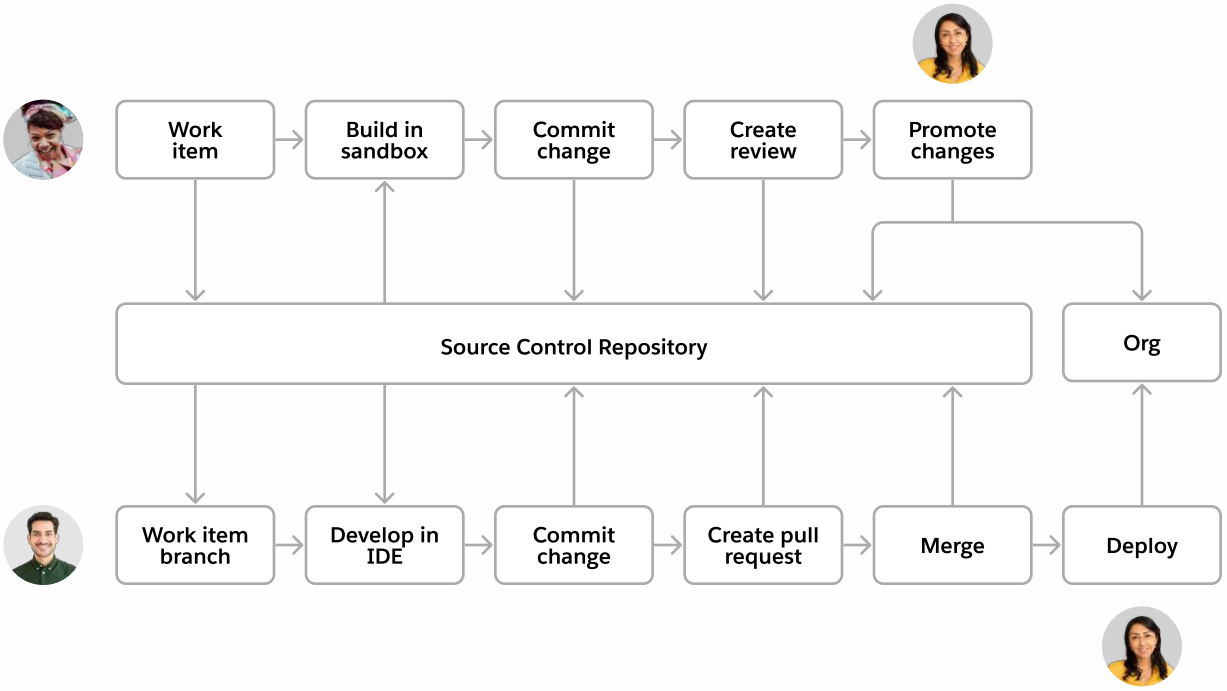
Pedro and DeeDee are working on the same feature – Pedro focusing on backend logic and DeeDee handles the UI components. Pedro works in a code editor with integrated command-line tools for code management, while DeeDee operates in a graphical interface. This allows both developers to collaborate seamlessly and keep their environment in sync. Pedro uses static code analysis and encryption early in the development cycle to prevent vulnerabilities.

Ramya
Release Manager

- Deploys changes using either declarative tools or command-line interfaces, ensuring that security policies (including automated vulnerability scans and compliance checks) are enforced at each stage of the deployment pipeline.

Once Pedro and DeeDee's changes are reviewed and approved, Ramya oversees a secure, compliant deployment to shared test environments. Using a modern DevSecOps platform, she can validate security policies and governance controls before promoting updates with just a few clicks.

Fusion Team Collaboration with DevSecOps



DevSecOps will help alleviate conflicts that often arise when multiple developers work on the same component in different environments. It can help keep multiple testing environments in synchronization.

Section 4

Five Stages of DevSecOps and Agent and Application Lifecycle Management with the Salesforce Platform

The Salesforce Platform helps teams build and customize secure AI agents, applications, and automations for both employees and customers. Meanwhile, a structured approach to agent and application lifecycle management (ALM) and DevSecOps make this possible.

Providing developers with the tools they need at every stage of the development lifecycle, these include open-source development supported by an active community, secure and isolated environments for testing, and built-in technologies to ensure data compliance and least-privileged access (to name a few).

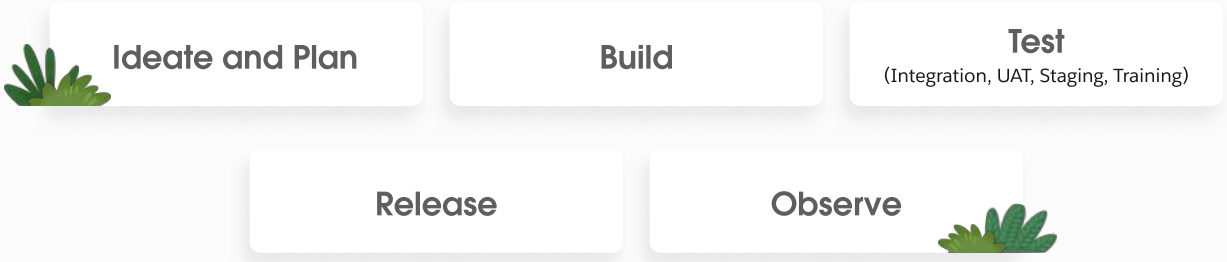
With modern development tools – such as scriptable CLI and open-source extensions for popular editors like Visual Studio Code – teams can work with their preferred setups and focus on high-value, non-repetitive tasks. These solutions help development teams and IT leaders secure their lifecycle while managing changes efficiently, reducing the costs associated with unclear and unsecured development pipelines.

To see how these tools fit together and how you can apply DevSecOps practices within Salesforce, let's first take a look at what an ideal Salesforce ALM looks like.

What is Agent and Application Lifecycle Management?

Agent and application lifecycle management (ALM) covers the full journey of an agent or application – from design to release. In Salesforce development, think of ALM as “what” you do (such as user acceptance testing), while DevSecOps practices are “how” you do it (like automating testing through continuous integration pipelines). ALM provides a blueprint to agile development, helping teams to move quickly, adapt to changes, and prevent costly production errors.

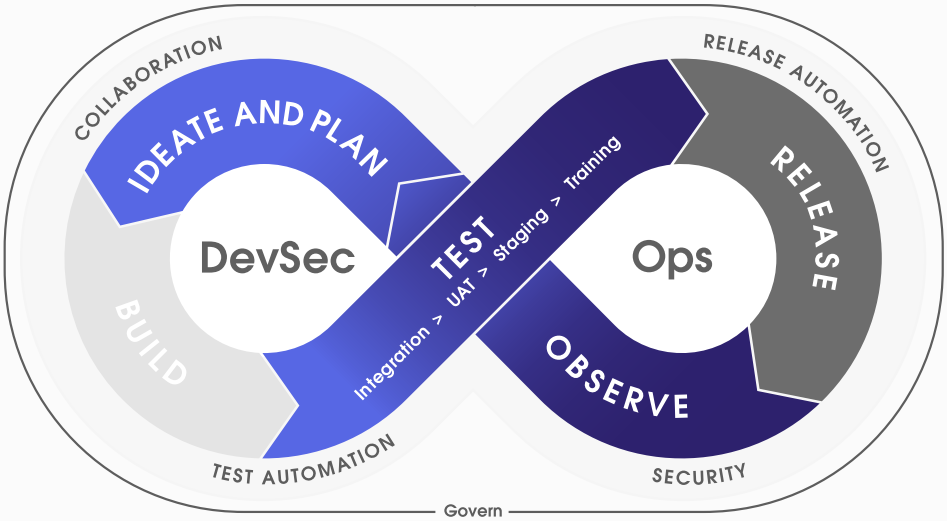
The five application lifecycle stages are:



Combining ALM with DevSecOps means you get the best of both worlds – teams can achieve faster and more reliable releases. This approach helps maintain code integrity, reduces errors, and makes collaboration easier across teams. Plus, with our open ecosystem of CI/CD partners, you can bring the tools you're already using and enjoy automated testing at every step of the ALM process.

The Salesforce Platform supports ALM and DevSecOps best practices, ensuring traceability and compliance. We recommend using source control as your single source of truth and testing on commit to keep everything running smoothly.

The ALM Process



Let's take a closer look at each stage of the ALM process and what it entails.

ALM Phase 1: Ideate and Plan

What to do

Before any code is written, it's important to map out the project with security from the start. That means defining requirements, breaking down tasks, and aligning on design specifications with your development and security teams. Think beyond just features – consider data protection, data classification, compliance needs, and potential risks upfront. This is also the time to determine which development and testing environments your teams will need as the project moves through the ALM cycle. Equally important is identifying the security measures that will be required to protect your data and ensure compliance throughout the process.

Tools to use

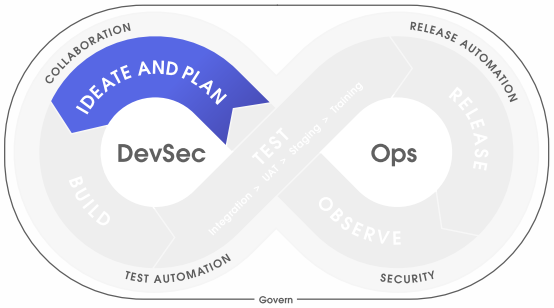
DevOps Center, Policy Center, Security Center 2.0, and Data Residency

Roles involved

Product Manager, Team Manager, Developer, Admin, Security Engineer

Use case

A project manager for your team's Salesforce instance collects feature requests and documents user stories in their planning tool – whether it's Jira, directly in Salesforce with DevOps Center, or Agile Accelerator. But planning isn't just about functionality. Security engineers work alongside developers early on to identify risks, define security controls, and ensure compliance requirements are met before development begins. By embedding security into planning, teams set themselves up for faster, safer, and smoother development down the road.



ALM Phase 2: Build

What to do

With a solid plan in place, it's time to start building. Developers and admins bring the design specifications to life, implementing features and customizations that meet business needs. But speed shouldn't compromise security. Every change should be built in a reliable environment that mirrors the latest source of truth. Security best practices, like static code analysis and encryption, should be embedded from the start.

Tools to use

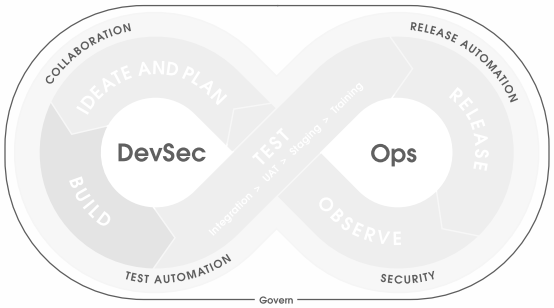
Sandboxes, Data Mask & Seed, Low Code Builders (for Agents, Prompts, Flows, and Apps), Agentforce for Developers, DX Inspector, IDEs, and more.

Roles involved

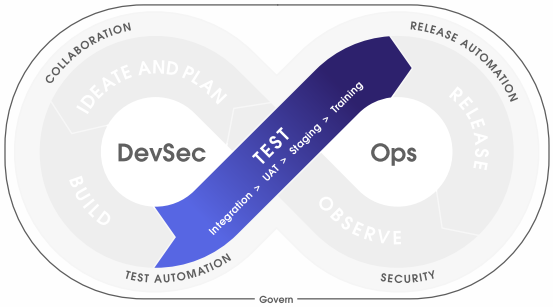
Developer, Admin, Security Engineer

Use case

Depending on the project's scope and complexity, developers or admins use pro-code or no/low-code tools to implement functionality. With security built in, Code Analyzer catches vulnerabilities early, while Platform Encryption protects sensitive data. For consistency, teams work in dedicated Developer Sandboxes, easily created or cloned using Hyperforce's Quick Clone and Quick Create. Developers can also use Salesforce



ALM Phase 3: Test



A. Integration

What to do

Testing isn't just about making sure things work – it's about making sure they work together. Integration testing helps catch conflicts as team changes converge, and using realistic, refreshed data helps surface issues synthetic environments might miss. While testing can begin in the dev environment, it's important to keep dev and integration environments separate to avoid cross-contamination. At this stage, teams consolidate all assets into a single release artifact and shift focus from individual changes to the release as a whole.

Roles involved

QA Engineer, Developers, Admin

Use case

As multiple developers and admins contribute changes, everything is merged into a Quality Assurance (QA) Partial Copy Sandbox. Automated testing ensures updates function correctly, while functional and scale testing help confirm performance. Developers, Admins, and QA engineers collaborate to verify that every change is secure, stable, and ready for the next phase.

B. UAT and Staging

What to do

User acceptance testing (UAT) makes sure that what you're about to deploy actually works for its intended users. The best way to do this? Test in an environment that mirrors production as closely as possible. A Full Sandbox provides an exact copy of your production org, complete with real-world data that has sensitive data masked and anonymized, so you can validate functionality in a setting that reflects how your users will experience it. Connect external systems to mirror production integrations and verify everything meets business and security requirements before moving to staging.

Tools to use

Agentforce Testing Center, Apex Testing & Debugging Tools, Low-Code Testing Tools, Sandboxes, Data Mask & Seed, Shield 2.0, Security Center 2.0, and Scale Test

Roles involved

QA Engineer, Business Stakeholder, Developer, Admin

Use case

Once testing in UAT is complete, the release artifact moves their changes to the UAT Full Sandbox, where teams validate UI, schema, and integration changes against full production data. Scale Test confirms that the application can handle peak traffic without breaking. And to ensure compliance, Data Mask & Seed secures sensitive information, allowing contractors and testers to work without accessing PII or PCI data.

C. Training

What to do

Rolling out a new app, agent or feature? A smooth transition depends on proper training – without putting live data at risk. Training environments allow users to explore new functionality, practice workflows, and troubleshoot potential issues before changes hit production. This eliminates confusion, reduces support tickets, and provides a seamless user experience.

Tools to use

Sandboxes, Data Mask & Seed

Roles involved

Business Stakeholder, Trainee

Use case

As teams introduce new custom app, agent, or feature, AgentExchange solutions, or newly released functionality, end users can train in a risk-free Full Sandbox environment that mirrors production. This allows them to test real-life scenarios without adding junk data to the live system. By the time the update rolls out, users are confident and ready to hit the ground running.

ALM Phase 4: Release

What to do

Once testing is complete and quality benchmarks are met, it's time to go live. But a smooth and secure deployment isn't just about pushing code – it's about making sure everything transitions seamlessly without breaking critical processes or introducing security gaps.

Tools to use

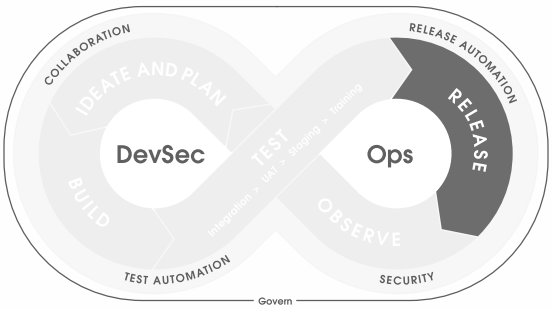
DevOps Center, Salesforce CLI, Hyperforce, Code Analyzer, Sandboxes, Security Audit, Shield: Event Monitoring, Security Center 2.0, and Backup & Recover

Roles involved

Release Manager, Product Manager, Team Manager, Business Stakeholder

Use case

The product owner and a select group of early adopters validate the release before it goes live. Once approved, updates are deployed to production, ensuring a secure, controlled rollout. If the feature is meant for external use, it can be distributed via AppExchange. After deployment, smoke tests confirm that core functions are running smoothly, and Backup & Recover secures a final snapshot of the data – because a great release isn't just about speed, it's also about stability and security.



ALM Phase 5: Observe

What to do

Monitoring your apps in production is key to keeping things reliable, secure, and running smoothly. The goal is to avoid downtime and stay ahead of security issues, so you catch problems before they affect users. Plus, it's a great chance to spot areas for improvement, which can go straight back into the planning phase to keep things improving over time.

Tools to use

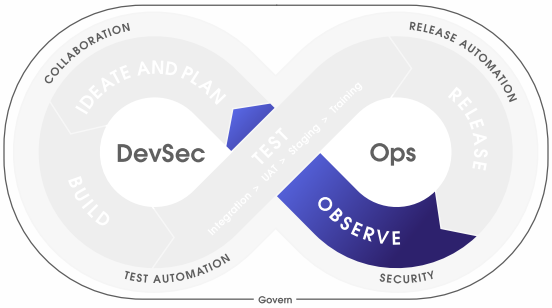
Event Monitoring, Salesforce Discover, Shield: Data Detect, Privacy Center, Security Center 2.0, Field Audit Trail, Scale Center, Digital Wallet, and more.

Roles involved

Product Owner

Use case

Once your application is live, real-time event monitoring gives you a window into who's accessing critical business data, when, and from where. You can also analyze user behavior to spot adoption bottlenecks and fine-tune performance. Monitoring tools like Scale Center track application performance post-release, while ApexGuru offers code recommendations to optimize scalability. Event Monitoring allows you to monitor user behavior, proactively find security threats, and respond to those threats quickly, strengthening security across your organization.



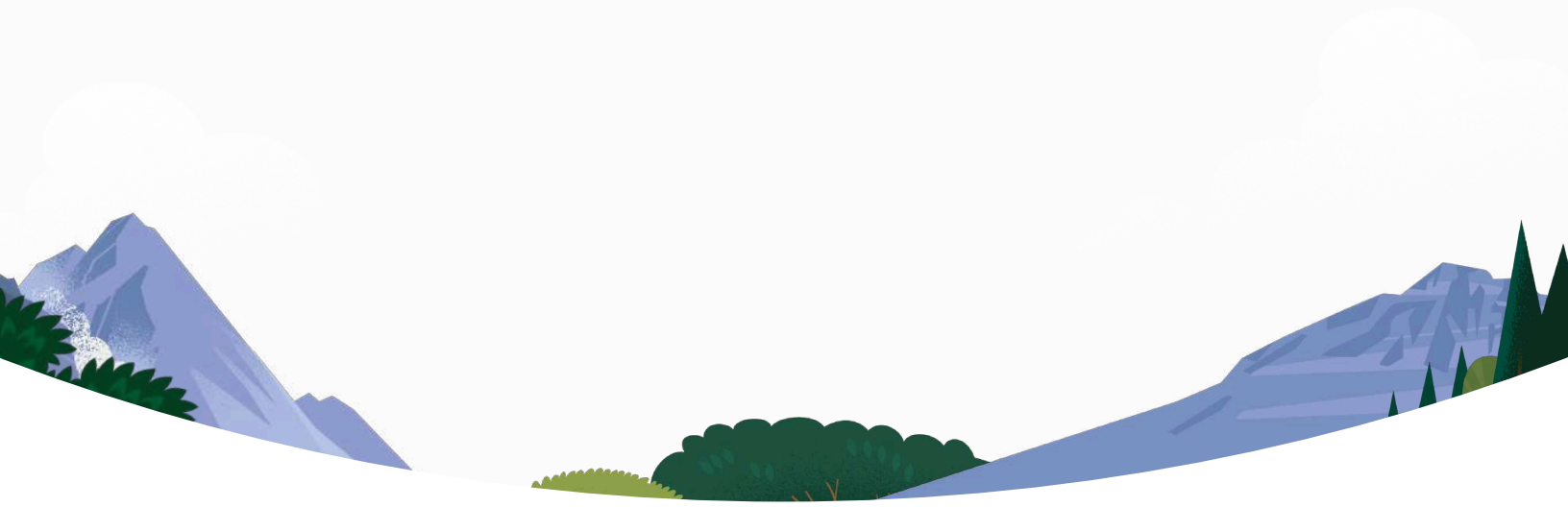
Governance across the ALM process

Governance isn't a single step – it's a continuous thread that runs through every phase of the ALM process. From ideation to post-deployment monitoring, organizations need guardrails in place to ensure compliance, data protection, and operational integrity at every stage. That means establishing clear security policies, enforcing them automatically through tools and controls, and maintaining visibility into who's accessing what, when, and why.

Governance also includes managing permissions with least privilege access, tracking data lineage, encrypting sensitive information, and preserving audit trails to support compliance readiness. By embedding throughout the ALM process and maintaining them consistently, teams can move quickly while ensuring trust, meeting regulatory requirements, and minimizing risk.

Tools to use

Shield 2.0, Archive, Backup & Recover, Privacy Center, Security Center 2.0, Data Mask & Seed, Hyperforce, Trust Layer, and more.



Section 5

Building Trust Through DevSecOps

In today's world of Agentforce and AI app development, security isn't just a checkpoint – it's a driving force. Waiting to patch up vulnerabilities after the fact? That's like locking the door after the burglars have left. Instead, security needs to be integrated into every step, making sure no process, data, or app is left unprotected.

That's where DevSecOps shines – when you shift security left in the development process, you ensure vulnerabilities are caught early, reducing risk and improving the overall security posture of your apps from the start. Because when security is built in, trust isn't just earned – it's a given.

Resources

Ideate and Plan

- Dive into DevSecOps
- Discover Agentforce
- Learn more about DevOps Center on Trailhead
- Check out Data Cloud

Build

- Learn about Agentforce for Developers
- Check out App Builder and Agent Builder
- Get the Data Mask & Seed datasheet
- Discover MuleSoft Topic Center

Test

- Learn more about Sandboxes
- Check out Agentforce Testing Center
- Discover Scale Test

Deploy

- Watch the DevOps Center demo
- Check out the Sandbox demo
- Learn more about Salesforce CLI
- Get the Backup & Recover datasheet

Observe

- Get to know Salesforce Digital Wallet
- Learn more about Shield
- Check out Salesforce Discover

Learn about Salesforce DevSecOps.

Visit our Testing and Deployment web page today.

Contact us to learn more: 1-800-667-6389