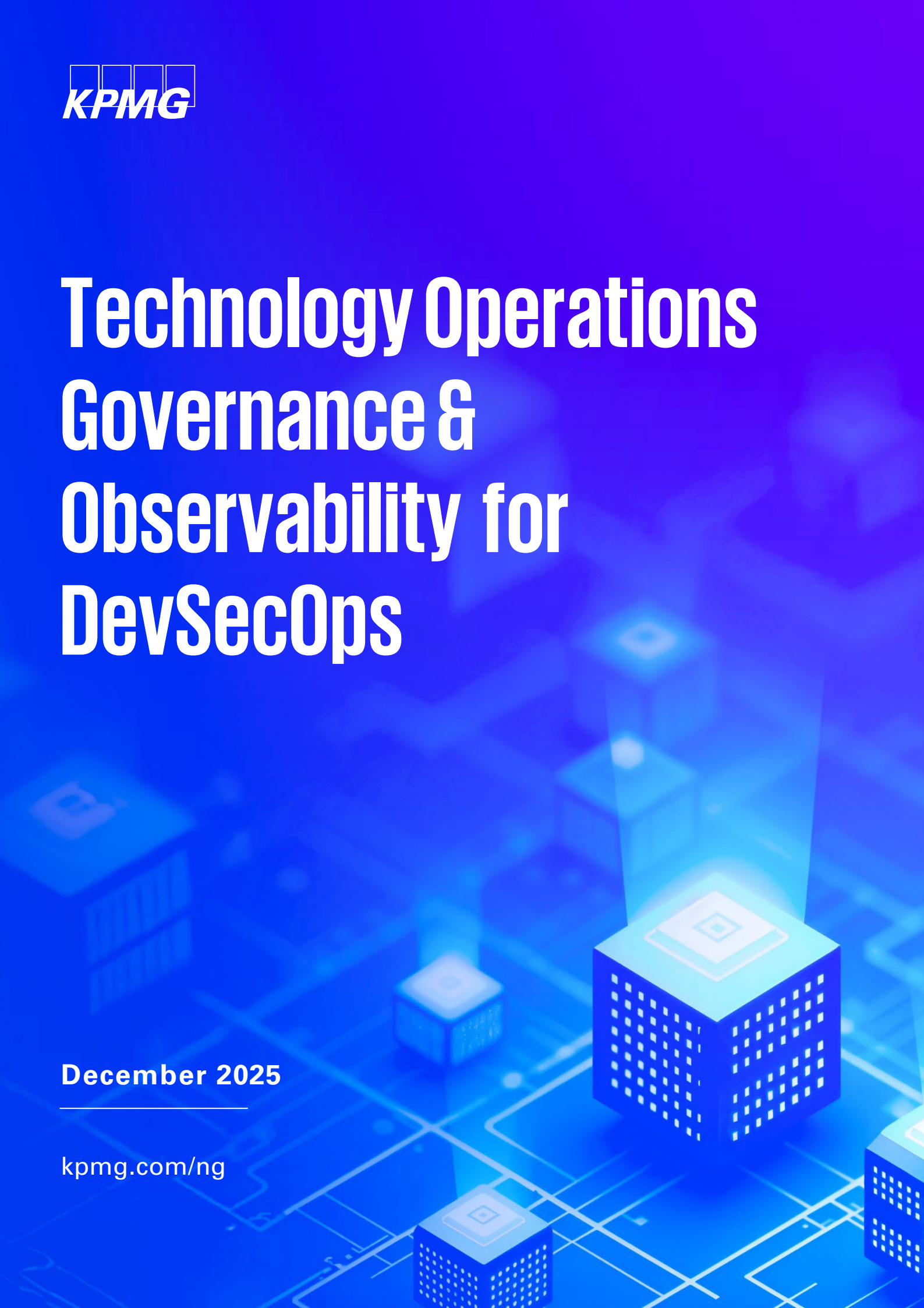
KPMG

Technology Operations Governance & Observability for DevSecOps



December 2025

kpmg.com/ng

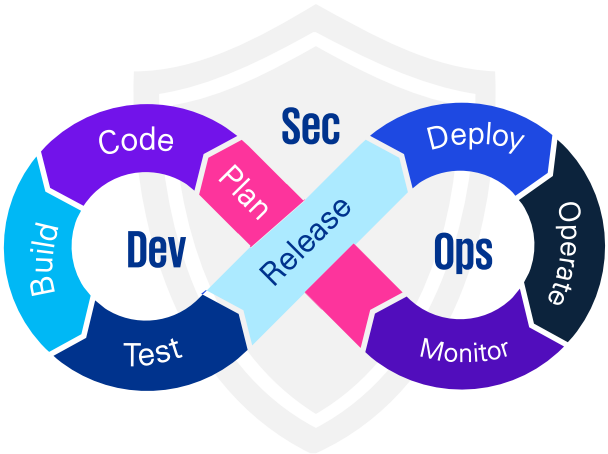
Contents

Introduction

Organizations are rapidly evolving from traditional development models to DevOps and, crucially, DevSecOps, embedding security proactively from design through deployment to mitigate inherent vulnerabilities often overlooked in the pursuit of speed.

While this integrated approach fosters agility, it also introduces layers of complexity, making observability essential: it's no longer just about collecting logs, but transforming them into a clear narrative of system health, performance, and security posture. This dynamic environment necessitates robust governance, well defined policies and frameworks to ensure controlled, compliant and value-maximized operations.

The truly transformative potential comes from observability and governance working together, continuously improving each other to create a resilient system that stays secure, adaptable and proactive.



…truly transformative potential lies in a symbiotic relationship where observability continuously informs and refines governance

Did you know?

A U.S. financial services firm lost over $440 million in 45 minutes due to a faulty code deployment, underscoring the need for real-time monitoring and strict deployment controls to prevent such catastrophic failures.

Governance in DevSecOps

As DevSecOps adoption grows, organizations must guide its implementation to capture real value. Without governance, efforts risk adding complexity without improving security, efficiency, or agility. Governance provides leadership with proactive control, helping standardize practices, avoid crises, and quickly align even minimal structures to proven best practices.

The focus here is not on institutional policies in the abstract, as the dynamic nature of technology often suffers under excessive "policing." Rather, it is the recognition that to truly realize the benefits of DevSecOps, it is equally important to establish guardrails, leveraging the very same technological tools that power modern delivery pipelines.

Why is effective Governance in DevSecOps especially urgent today?

1. **Complex Operating Environments**
   Distributed microservices, multi-cloud platforms, and hybrid IT ecosystems demand coordinated oversight.
2. **Escalating Regulatory Expectations**
   Regulators worldwide are increasing scrutiny over data privacy, operational resilience, and AI usage.
3. **Heightened Cybersecurity Risks**
   Automated pipelines can be weaponized if not governed, turning speed into a liability..
4. **Boardroom Visibility**
   Operational resilience is now a board-level priority, with stakeholders demanding evidence of robust governance..

We will explore the specific technology guardrails necessary to help organizations maximize the value of their technology operations. These examples are by no means exhaustive, but their tangible benefits underscore the necessity of proactively implementing governance tools and controls across the enterprise.

Governance Practices in DevSecOps

Static Application Security Testing (SAST)

Automated scans of source code flag insecure libraries, coding flaws, and outdated dependencies before deployment. Integrated into CI/CD, SAST supports early remediation and reduces the cost of fixing defects post-release.

Configuration & Compliance Checks

System and network configurations are continuously validated against industry benchmarks (CIS, NIST, PCI). Automated scans catch drift or misconfigurations early, ensuring environments remain secure and compliant.

Secrets Management & Leak Detection

Sensitive credentials, keys, and certificates are stored in secure vaults, rotated regularly, and kept out of code. Automated leak detection scans catch exposures in real time, reducing the risk of unauthorized access.

Infrastructure-as-Code (IaC) Scanning

Sensitive credentials, keys, and certificates are stored in secure vaults, rotated regularly, and kept out of code. Automated leak detection scans catch exposures in real time, reducing the risk of unauthorized access.

Operational Resilience Standards

System and network configurations are continuously validated against industry benchmarks (CIS, NIST, PCI). Automated scans catch drift or misconfigurations early, keeping environments secure and aligned with standards.

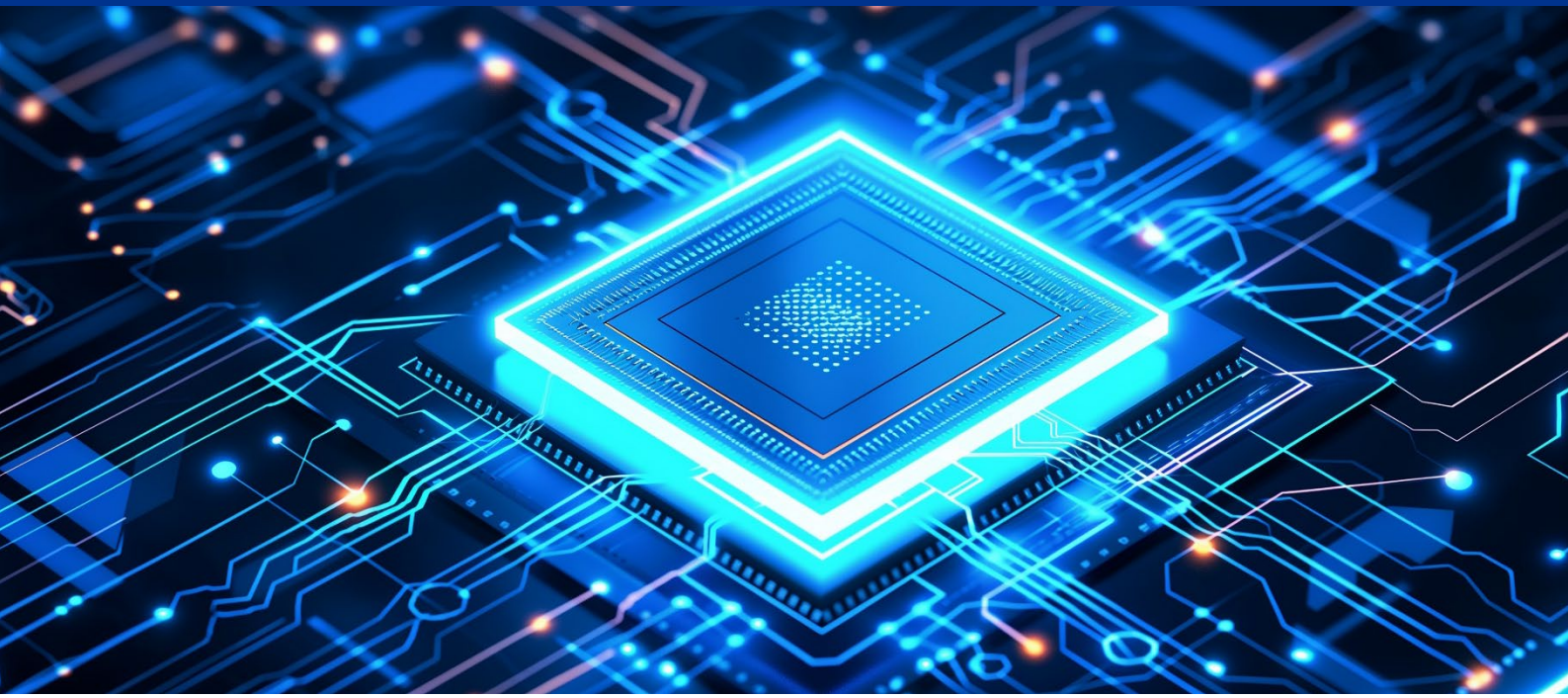
Policy-as-Code & Governance Automation

Policies for security, compliance, and risk management are codified and enforced automatically within CI/CD pipelines. This ensures that every deployment is checked against governance rules, reducing reliance on manual oversight and providing a verifiable audit trail of policy adherence.

Here is an insight: Shift-left in DevSecOps emphasizes embedding governance controls such as security checks, compliance validation, and policy enforcement earlier in the development lifecycle. By integrating these preventive measures from the planning and design phases, teams ensure that code, infrastructure, and processes align with organizational and regulatory requirements before deployment

Risks Mitigated Through Governance

Risks Addressed	Description
Regulatory and Compliance Risk	Regulatory bodies are increasingly vigilant about operational resilience, cybersecurity, and data protection. Without proper governance, organizations risk non-compliance with frameworks such as NDPR, GDPR, PCI DSS, HIPAA, and emerging AI regulations. Governance ensures continuous compliance by codifying requirements into daily workflows rather than treating them as periodic checklists.
Technology Costs	Without governance, DevSecOps teams risk proliferating tools, platforms, and architectures with little coordination. Over time, this creates inefficiency, duplication, and hidden costs. Governance addresses this by enforcing architectural standards, mandating lifecycle management, and ensuring technology investments remain aligned to enterprise strategy.
Reputational & Strategic Risk	Failures in operational governance manifest quickly as reputational damage and regulatory scrutiny. A single uncontrolled outage or breach can erode stakeholder confidence and invite costly oversight. Effective governance demonstrates to boards, regulators, and customers that technology operations are well controlled, balancing innovation with assurance.
Security and Data Protection Risk	A DevSecOps pipeline without governance is vulnerable to malicious code injections, privilege misuse, and unpatched vulnerabilities. Governance practices—such as mandatory code scanning, vulnerability management, and zero-trust access controls—address these risks directly.
Operational and Service Reliability Risk	Unchecked changes and uncontrolled deployments are a recipe for outages. Governance introduces structured guardrails—such as deployment approvals, SLO monitoring, and rollback protocols—that mitigate reliability risks. By embedding resilience into governance, organizations reduce the likelihood of SLA breaches and unplanned downtime..



Observability in DevSecOps

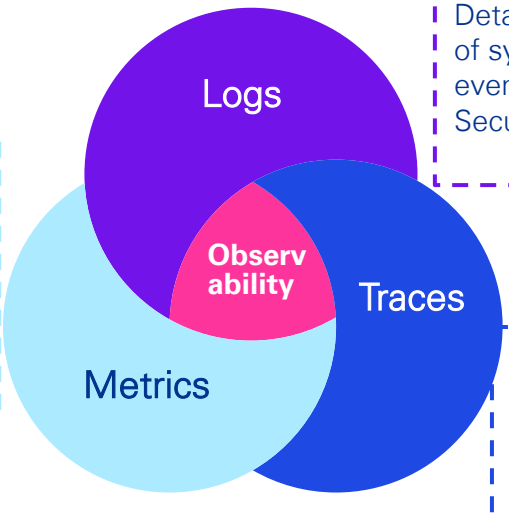
While Governance in DevSecOps sets the rules and guardrails for secure software delivery, **Observability ensures the visibility needed to enforce and validate them in real time**. It provides continuous insight into system performance, security posture, and compliance across the development pipeline, enabling teams to detect anomalies, trace root causes, and respond quickly.

Observability in this context goes beyond traditional monitoring, which only tracks predefined metrics, logs, or alerts when something breaks. Instead, it analyzes logs, metrics, traces, and security signals in a unified way to explain why a system is behaving a certain way. This not only helps detect known issues but also uncovers previously unseen problems. For example, while monitoring might raise an alert for high API latency, observability can trace the request across services and reveal that the slowdown is caused by a misconfigured firewall rule combined with a vulnerable dependency.

The building blocks that make Observability possible are logs, metrics, and traces. Together, they provide DevSecOps teams with the visibility to detect issues, uncover risks, and maintain control across the delivery pipeline.

What happened?
Detailed, timestamped records of system and application events e.g., Application logs, Security logs, etc.

How is it Performing?
Numeric measurements that track the health and performance of systems e.g., latency, error rates, number of vulnerabilities per build, etc.



Where did it happen?
A record of the path a request or transaction takes as it flows through services e.g., CI/CD Pipeline traces, Application & Microservices traces, etc.

Observability Practices in DevSecOps

Vulnerability Scanning

Employing the use of automated vulnerability scanning on all assets to find security issues like outdated patches and open ports. Prioritizing findings using Common Vulnerabilities and Exposures / Common Vulnerability Scoring System (CVE/CVSS) standards and integrating them into automated workflows for patching and remediation to maintain a secure environment.

Auditability & Traceability

End-to-end traceability through immutable logs and versioned change histories makes every action visible and verifiable. Automated evidence collection ensures accountability and simplifies audits or incident reviews.

Container Image Scanning

Using automated container image scanning to detect vulnerabilities and malware before deployment. Policies can be set to automatically block or roll back high-risk images based on Common Vulnerabilities and Exposures (CVE) severity, preventing insecure containers from entering the production environment.

Scaling and Resource Allocation

Observability helps teams make informed decisions about their use, improvement and allocation of technology resources by analyzing trends in resource usage which improves decision making.

Shift-Left Strategy

In DevSecOps practice, observability promotes early detection of issues which in turn is addressed head on during the early stages of the software development process. This improves collaboration and makes the end-to-end process of IT operations and security a shared responsibility amongst team.

Automated Incident Response

Data from observability is used to configure automated workflows and remediation steps through system triggers. This alerting system powered by observability is implemented and configured to prioritize issues based on severity and impact.

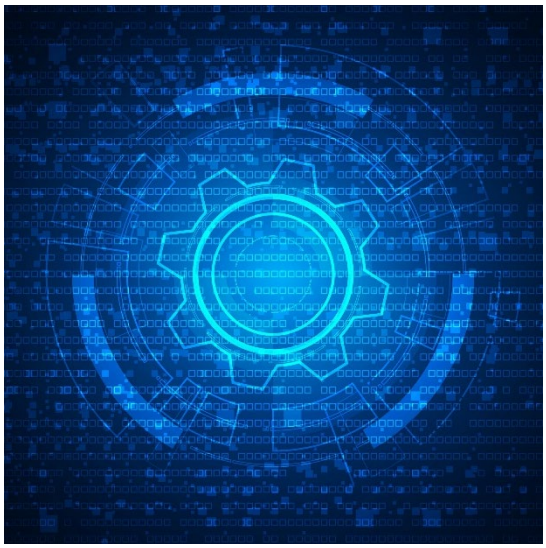
Here is another insight:

"You can't secure what you can't see". Observability goes beyond performance monitoring to uncover hidden blind spots where threats may lurk. It enables early detection, unified visibility, intelligent response, and continuous compliance.

Risks Mitigated Through Observability

Risks Addressed	Description
Engineering & DevOps Risks	Features are deployed without visibility, raising the risk of defects in production. These may lead to slower change releases as teams find it difficult to debug. Additionally, having fragmented tools may create blind spots and inconsistent insights. Observability gives real-time feedback on production behavior, enabling safer deployments, faster debugging, and unified monitoring
Operational Risks	Undetected runtime vulnerabilities in deployed containers, images, or application. Hidden issues like data corruption or degraded performance may go unnoticed until they cause outages. Observability provides real-time metrics, logs, and traces that surface hidden failures, speed up detection (lower Mean Time to Detect (MTTD)), and give engineers the context needed to diagnose and fix issues quickly (lower Mean Time to Respond/Repair/Recover (MTTR)).
Performance & Reliability Risks	Applications may be slow or unstable, and without proper tracing/metrics, the cause is invisible. Additionally, scalability maybe threatened as teams risk overloading systems without visibility into resource consumption trends. Observability helps detect subtle degradations (latency, partial failures) before they escalate to outages.
Security and Compliance Risk	Lack of telemetry may let attackers move laterally or exfiltrate data unnoticed. Many regulatory frameworks (e.g., ISO 27001, SOC 2) require logging, monitoring, and auditability. Without logs/traces, you can't detect unauthorized or anomalous behavior. Essentially, poor observability practices lead to compliance issues. Observability ensures comprehensive logging, monitoring, and tracing that detect suspicious activity, provide evidence for compliance, and help identify insider misuse early.
Business & Customer Risks	Outages or performance issues reduce user trust and satisfaction translating to poor customer experience. This may lead to revenue loss in digital businesses. These unexplained failures harm brand reliability and stakeholder confidence. Observability provides end-to-end visibility and customer-impact insights, helping organizations detect and resolve issues faster, minimize downtime, and communicate transparently.

Governance and Observability: The Backbone of Secure Development Operations



Integrating Governance and Observability into the DevSecOps framework is essential for strengthening both security and operational efficiency in today's fast-paced digital landscape. Governance ensures that development activities align with organizational policies, industry standards, and regulatory requirements, while Observability provides real-time visibility into system performance, risks, and compliance posture.

Together, they enable proactive risk management by detecting vulnerabilities and compliance gaps early in the development cycle, reducing the likelihood of costly incidents or regulatory breaches. They also simplify oversight by consolidating controls and monitoring into a unified framework, which reduces complexity and enhances accountability.

For leadership, the business value is clear: faster, more secure delivery of digital solutions, reduced operational risk, improved stakeholder trust, and sustained compliance with evolving regulations. Ultimately, embedding Governance and Observability in DevSecOps ensures innovation can scale without sacrificing security or compliance.

Use Cases (+)

Case A: DevSecOps Transformation in a Financial Institution

In 2023, Bank A took on a large-scale DevSecOps transformation project to modernize its technology practices and stay competitive in a fast-moving financial sector. The initiative was driven by a comprehensive review of existing workflows, which revealed inefficiencies in deployment, testing, and system management. With these insight the bank, repositioned technology as a catalyst for agility and resilience, aiming to bring teams together, make better use of resources, and deliver faster, more reliable services to customers.

Approach

Bank A focused on embedding DevSecOps as a core business process by implementing: real-time monitoring for proactive error detection, automating testing to reduce pipeline bottlenecks, and deploying containerized environments to strengthen reliability and simplify application updates.

Outcome

DevSecOps adoption strengthened cross-functional collaboration by unifying workflows across departments, resulting in improved project alignment. Operational efficiency also increased, with automation reducing manual interventions by **40%**, accelerating deployment cycles, and minimizing error rates.

Case B: DevSecOps in a Telecommunications Company

A leading telecommunications company, long challenged by regulatory compliance requirements, undertook a DevSecOps transformation to integrate security directly into its development and operations processes. Prior to this, compliance-related issues often slowed product releases and strained coordination between teams. By adopting DevSecOps, the company aimed to streamline workflows, integrate controls early, and strengthen accountability across functions.

Approach

The company integrated **continuous integration and continuous deployment (CI/CD) pipelines**, ensuring every code change was automatically tested against security policies. This fostered proactive collaboration across development, operations, and security teams, embedding compliance into the software delivery lifecycle.

Outcome

By embedding security into their workflows, the organization achieved a **30%** reduction in compliance-related issues. This demonstrated how effective cross-team communication and automated controls can drive successful DevSecOps adoption, while also enhancing resilience and regulatory alignment.

Use Cases (-)

Case A: Misconfiguration in a Cybersecurity Provider's DevOps Environment

A global networking and cybersecurity provider, suffered a massive breach involving the theft of 4.5 terabytes of sensitive data. The attackers gained access through a misconfigured DevHub instance, bypassing security barriers and exposing weaknesses in the company's cloud and DevOps setup. The incident underscored how easily overlooked misconfigurations can escalate into large-scale compromises.

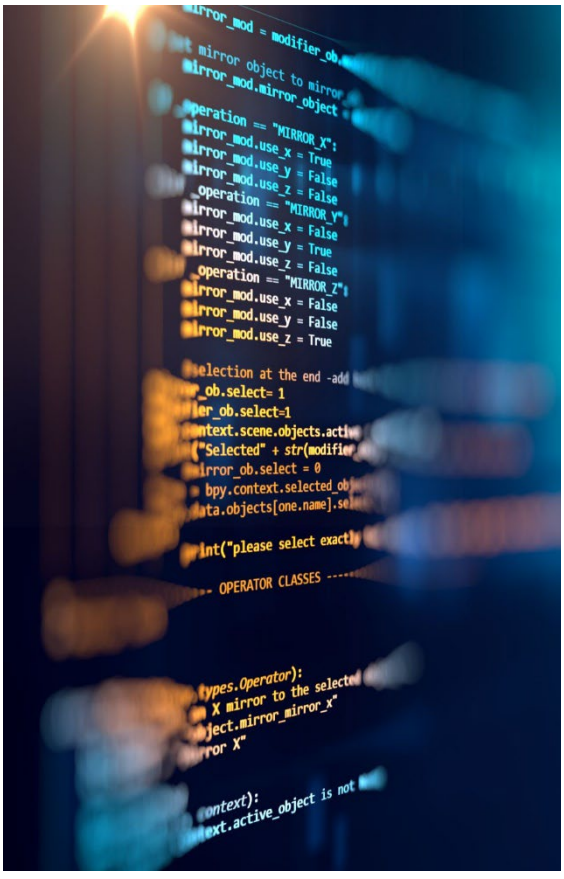
Shortfalls

- Misconfiguration of DevOps environment
- Poor Network segmentation

Critical Takeaway ⚠

DevOps environments must be treated as high-priority assets. Organizations should enforce strict configuration management, segment networks, and encrypt sensitive data. Clear incident response plans and ongoing employee training are also essential to minimize human error and strengthen security posture.

Case B: Unpatched Vulnerabilities in a Credit Reporting Agency



A major credit reporting agency, was breached when attackers exploited a known flaw in Apache Struts through its consumer complaint web portal. Due to poor patching practices and weak network controls, the attackers moved laterally, accessing multiple systems and extracting hundreds of millions of customer records. The breach went undetected for months, partly because a key security certificate had not been renewed.

Shortfalls

- Failure to patch critical Apache Struts vulnerability (CVE-2017-5638)
- Weak network segmentation enabling lateral movement
- Plaintext storage of credentials on servers

Critical Takeaway ⚠

Timely patch deployment is critical to reducing exposure windows. Continuous monitoring and certificate management help detect intrusions earlier. Sensitive data should be encrypted, access controls strictly enforced, and networks segmented to contain breaches.

How KPMG Tech Risk can Help

Complying with regulatory standards does not provide immunity nor does it adequately protect against cyberattacks. DevSecOps requires constant vigilance across the software development lifecycle, from secure coding to real-time threat detection, to truly protect your systems. KPMG Tech Risk is well positioned to provide the following services pertaining to the implementation of security measures within DevOps and the deployment DevSecOps.

DevSecOps Maturity & Risk Assessment: Our DevSecOps Maturity & Risk Assessment helps organizations evaluate how effectively security is integrated into their development and operations processes. Leveraging industry frameworks, we assess key areas such as automation, incident response, compliance automation, security testing, infrastructure-as-code and threat modeling.

Technology and Cybersecurity Internal Audit: We provide a structured, independent evaluation of your organization's IT infrastructure, systems, and security controls to ensure alignment with business objectives and regulatory requirements. Our audit delivers clear insights into digital resilience, identifies control gaps, and offers actionable recommendations to strengthen your organization.

Technology Governance Gap Assessment: We help you identify discrepancies between current and desired technology governance practices using frameworks like COBIT and ITIL. Identifying weaknesses in strategic alignment, risk management, and value delivery, enabling targeted improvements.

Secure Architecture and Tooling Design Review: We perform a structured evaluation of technical architecture and tooling to identify security weaknesses and ensure alignment with security requirements before development. We apply threat modeling to recommend security improvements.

Post Deployment/Implementation Security Review: We verify that security requirements and controls are properly implemented and functioning after deployment. Identifying configuration drift and vulnerabilities, providing feedback to strengthen the security lifecycle.

Contact us

Lawrence Amadi
Partner and Head
Tech Risk
KPMG West Africa
T: +234-803-975-4017
E: lawrence.amadi@ng.kpmg.com

Chukwuemeke Igabari
Associate Director
Tech Risk
KPMG West Africa
T: +234 702 500 1098
E: chukwuemeke.igabari@ng.kpmg.com

home.kpmg/ng
home.kpmg/socialmedia