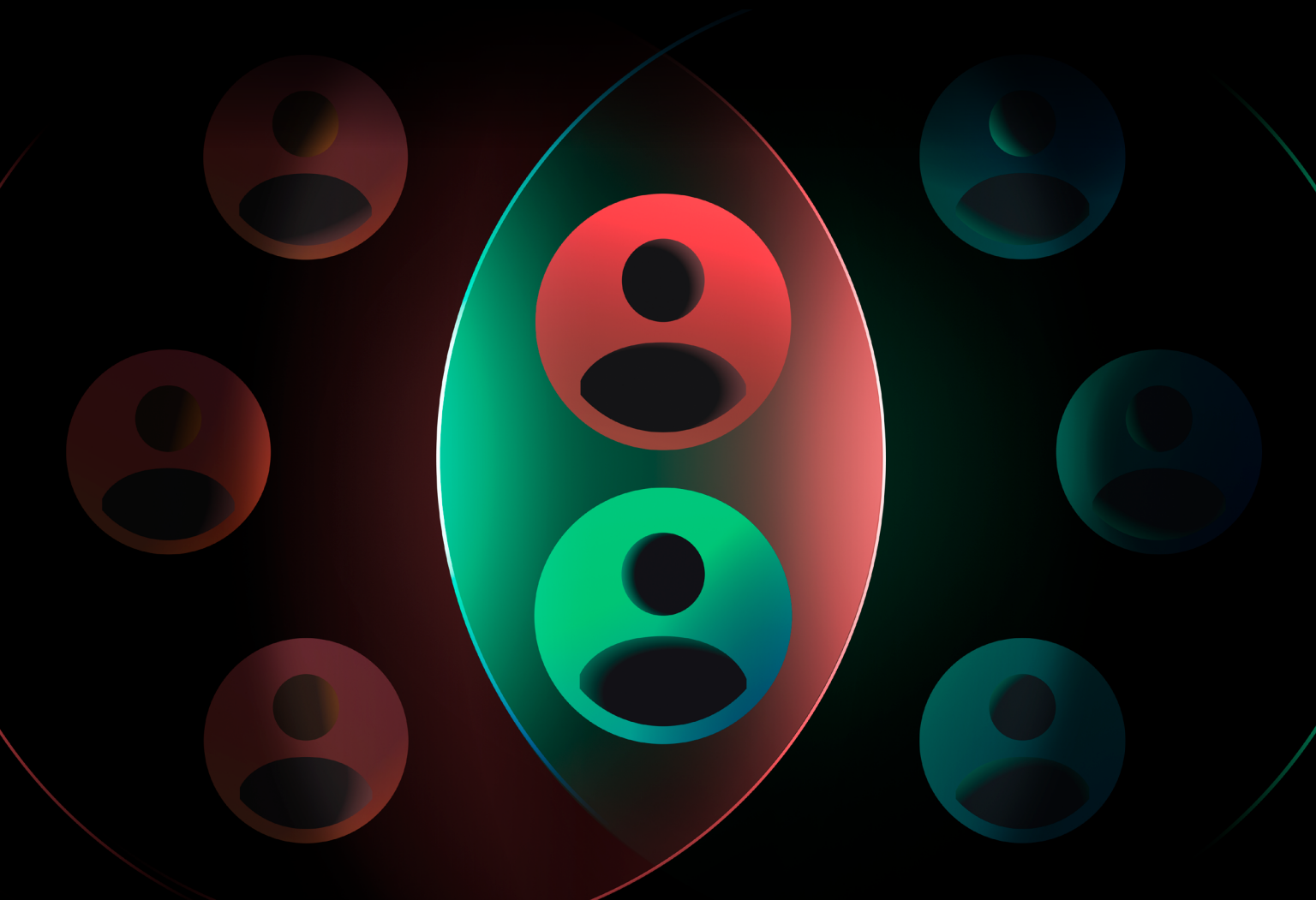
HashiCorp

E-BOOK

# Peace with platforms

## Improving the efficiency of developer and security teams



# Table of contents

# Introduction

Trust remains a key barrier to the cloud. Although 94% of organizations run workloads in public or private clouds, the cloud houses just 60% of enterprise data, and cloud services themselves remain a mixed bag from a software and data security standpoint.

Cloud security remains a work in progress:

- 91% of enterprises have experienced software supply chain incidents in the past year (**Enterprise Strategy Group**).
- 75% marks the rise in cloud incidents between 2023 and 2024 (**Crowdstrike**).
- 74% of all breaches include a human element — problems attributed to errors, privilege misuse, or stolen credentials, among other issues (**Verizon**).

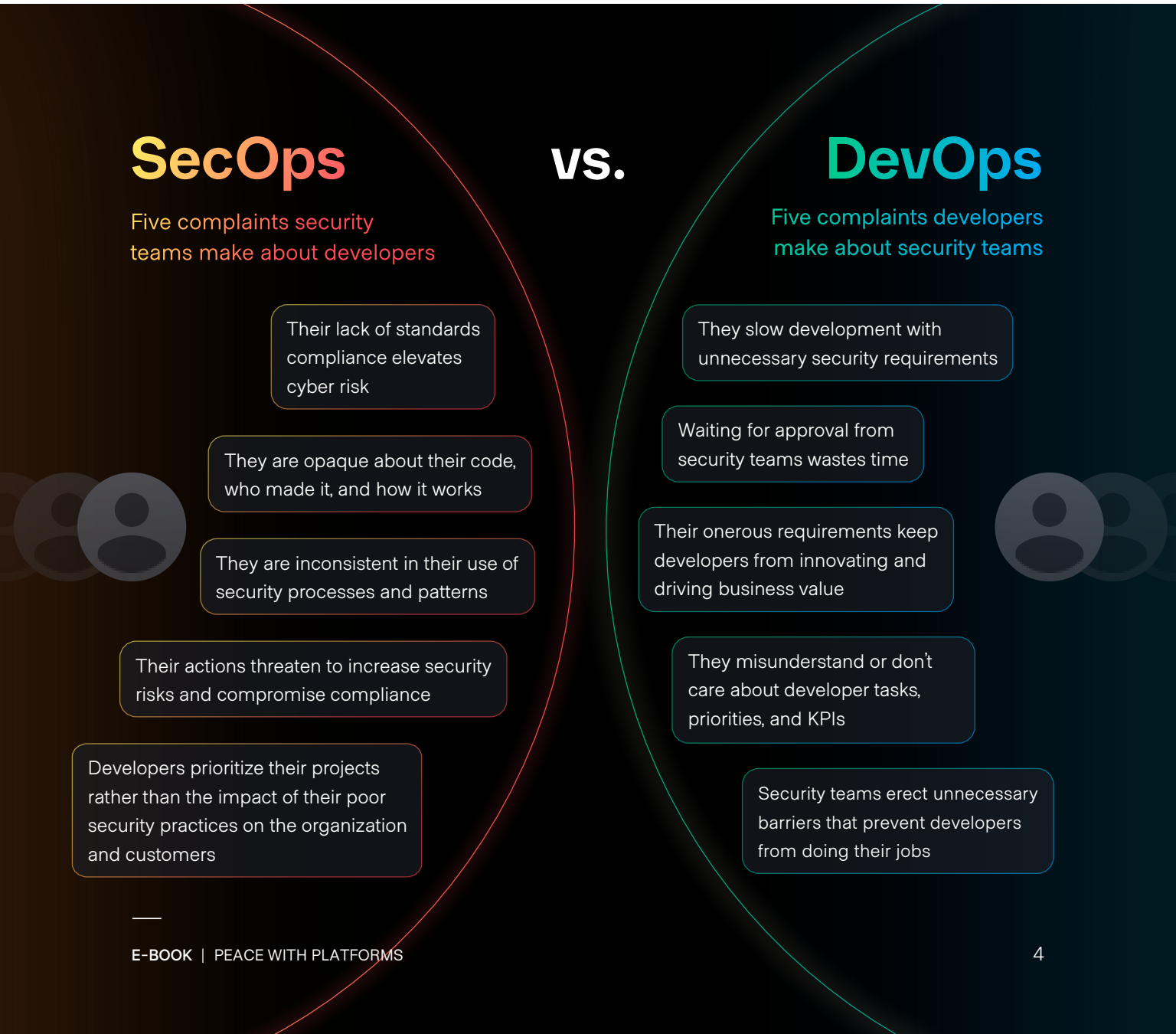
Those statistics tell a sobering story, but it doesn't capture all of the risk and complexity facing organizations that run workloads, use third-party APIs, or implement services in public, hybrid, or multi-cloud environments. There are many reasons why these problems persist. A lot has been written about the topic of cybersecurity, but this e-book spotlights one notably overlooked factor: **Misaligned security and development teams**. The hallmark of every misalignment is inconsistent goals and tooling between teams, ignored security practices, cloud misconfigurations, lapsed communication, and general friction.

Until now, this disharmony has received scant research. There's little formal data regarding this conflict or how its tensions predate the cloud. Yet, nearly two in three CISOs and developers agree that "a lack of communication and collaboration between their teams is a problem when it comes to implementing better software supply chain security," according to a **software security** study. The study identified another sensitive issue when it found that tooling spurs conflict between these teams: "73% of developers agree that the work/tools their security team requires them to use interferes with their productivity and innovation."

Why does this matter? Three reasons.

1. Prolonged tension between these teams does not enhance cloud security, and, if anything, it worsens matters: slowing development, adding costs, and heightening security risks.
2. The problem is fixable (and we'll explain how).
3. Getting this right matters.

In our **State of Cloud Strategy Survey**, four out of five decision makers (81%) indicated that security is fundamental to determining the success of their organization's cloud strategy. The study found that increased and improved automated tooling to manage the entire security lifecycle is essential to achieving key business goals. This e-book serves as a guide to help organizations identify and resolve this longstanding, corrosive behavior and accelerate their secure cloud development workflows.



## The TL;DR — Dev and Sec team troubles

Where did things go so wrong between these pivotal teams? Mark Zuckerberg famously coined the motto "Move fast and break things" in 2014. However, the behavior codified by the Facebook founder had long since been baked into the "hacker" culture. The Sec and Dev teams' issues were exacerbated in the hectic early days of the public cloud (circa 2006). Many organizations were racing to shift data, apps, and services to the cloud. However, security teams lacked adequate tools or processes, and it often took them much longer to review code than for developers to generate it.

Dev and Sec teams quickly fell out of sync production-wise. Finger-pointing ensued. The tension still exists privately on Slack but more publicly on Reddit.

> "You can send them [developers] all kinds of information on OWASP, DevSec etc, most of them won't care and will simply ignore it until management tells them to care."
> - **Reddit user**

Security teams often work under stressful conditions with time constraints. They face existential threats such as blame (or fines) for breaches or responsibility for regulatory noncompliance. There have been countless efforts by Sec teams to impose guardrails on Dev teams to help enforce desired security policies and best practices. These efforts have not been well-received or, more importantly, accepted by all developers.

For their part, developers often receive security training but don't want to spend much time thinking about it. They're generally not held to account for security vulnerabilities, and usually, there are no financial incentives or penalties at stake for them.

## The role of platform teams

**Platform teams**, often composed of cloud architects and engineers, occupy a unique and increasingly pivotal organizational role in deploying, securing, and managing applications. Platform teams become "plumbers" of the IT organization, because they help build a standard infrastructure workflow to underpin the entire IT estate. They may architect or choose the development tools and platforms that improve developer velocity — speeding up time-to-market by removing friction from the development process.

These plumbers aim to abstract away infrastructure complexity and achieve this objective by building or deploying infrastructure as code templates. For these and other reasons, platform teams play an

important, if underappreciated, role in resolving the longstanding impasse between Sec and Dev teams.

Yet, in the absence of a platform strategy, organizations have encountered issues such as:

- **Poor tool onboarding.** This problem happens when developers leave their comfort zone — igniting cultural, training, and skill gap issues.
- **Continuous delivery or compliance problems**. Organizations need help to balance speed and security, particularly in highly regulated industries.
- **Overwhelming complexity**. Mismatched security and development tools not built for multi-cloud or cloud native environments.

Platform teams' decisions and actions can overcome these challenges. In their crucial role as architects of a platform strategy, they will select the tools and development platforms to manage the entire Software Development Lifecycle (SDLC). Most critically, these teams must simultaneously prioritize cloud security and the developer experience, establishing a secure and consistent workflow that supports all teams in the delivery pipeline. Integrating security best practices into developer workflows (often called a "shift-left" approach) is critical to building more secure software without slowing development.

Eliminating slow, time-consuming, manual steps makes every team happy. Both groups want platform teams to implement more automated systems that require fewer manual approvals or ticketed workflows. How do the new workflows deliver sustainable value? Striking a balance between security and speed, the new platform workflows must:

- Accelerate developer velocity for launching new infrastructure securely.
- Implement a centralized secrets management platform.
- Secure keys and credentials that applications use with minimal developer friction.
- Put into place prebuilt security policies to prevent insecurities from making it into production.
- Provide faster access to a tightly defined list of privileged systems.
- Connect services quickly and securely over the network.
- Speed up debugging and auditing.

## Consistent tooling matters

Organizations with misaligned security and development teams face a broad spectrum of risks. They may incur direct costs through project delays and longer times to market. And security vulnerabilities from improperly built software can lead to common patch failures or more catastrophic, reputation-destroying breaches.

Even when it's unintentional, misaligned priorities, mismatched tools, and inconsistent workflows heighten friction or lead to poor collaboration between Sec and Dev teams. The tension doesn't go unnoticed. Cybersecurity training aims to build awareness and promote an understanding of security priorities — but mitigating threats is often the primary objective.

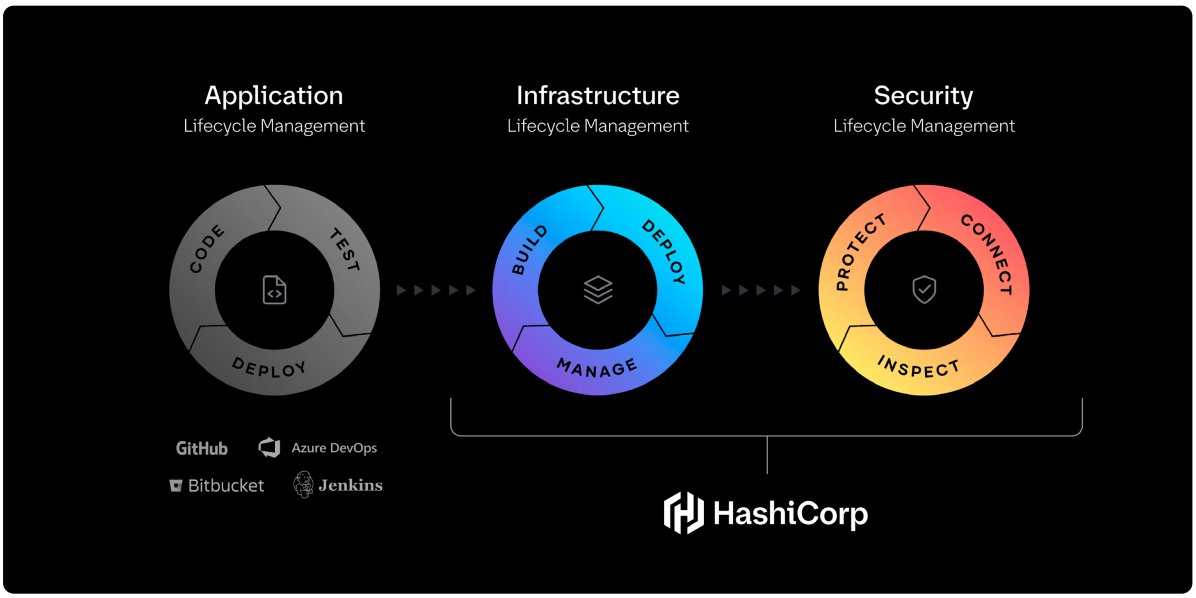
Although the danger may not seem imminent, reducing tool sprawl helps minimize the chances of misconfigurations or errors that lead to these problems. Yet tool sprawl, like misaligned teams, doesn't happen by design. Most security teams work with tools from preferred suppliers, tools they have grown comfortable using over many years. Ditto for developers.

> "97% of developers complain that they context switch because the tools are from multiple vendors. Because they don't work out of the box, you need to enable integrations. And it's not one central pane of glass, you need to switch interfaces. These metrics show us that too many tools negatively impact developer experience, which in turn decreases development."
> — **Harsha Vathsayayi, a product manager at Roche Informatics**

Platform teams understand that the right tools can propel a cultural shift — swinging the pendulum from team-centric to cloud-centric, organization-wide best practices. With tool and process consistency, you get smoother Dev and Sec workflows. Here are several examples of how this works in practice:

> "What typically used to happen was, developers don't read security standards. We get to the point where we've got the UAT (user acceptance test), and that's where we have our pentest (penetration test), and so on. Then the developer gets told they've breached some standards, so they have to go back, rebase the code, resubmit it through, and they miss the pentest window. It costs Barclays money. Bringing that failure as close as we can to the development stage means that we avoid all of that unnecessary extra cost."
> — **Using Terraform Enterprise and Chef to enable continuous deployment at Barclays**

> "On every pull request, a plan is automatically triggered. We will immediately see if the plan succeeds or if the plan fails. We have now the proactive part for our policies. All our departments, like governance and security and our central platform team, can now write policies as code that define what is allowed and what isn't. All users immediately see if their code is compliant or not. Also included is cost estimation."
> — MediaMarkt's journey to compliance with Terraform




## Getting lifecycle management right

Platform teams focus on end-to-end lifecycle management, including application lifecycle management, which establishes the workflows, systems, and capabilities to code, test, and deploy applications to cloud infrastructure.

They also manage the lifecycles that intersect developers and security teams — creating:

- Infrastructure Lifecycle Management (ILM), which establishes a systematic and repeatable approach to creating, securing, and maintaining infrastructure. It includes the code to build, deploy, and manage the infrastructure that underpins cloud applications.

- **Security Lifecycle Management** (SLM), which provides a systematic way for organizations to manage their most sensitive data, especially secrets/credentials, from creation to expiration or revocation. It uses identity-based access controls to manage the security lifecycle of secrets, user access, and services.

That's the objective of well-managed security and infrastructure lifecycles. Mastering SLM and ILM enables organizations to slash risk, lower costs, accelerate developer velocity, and secure their most critical apps while reducing tension between Dev and Sec teams.

Simplifying cloud development is crucial to achieving compliance and cloud security. According to IBM's **State of the Cloud** report, 53% of firms agree that ensuring compliance in the cloud is too difficult. And among those firms, four in five (82%) agree that their team is "lacking skills necessary for them to be proficient at architecting and/or managing cloud applications." Shifting left reduces team stress and enhances productivity. Simplifying the development process and aligning Dev and Sec toolchains enables organizations to accelerate delivery and improve compliance and cloud security.

**The Infrastructure Cloud**, powered by the HashiCorp Cloud Platform (HCP), helps developers work faster by turning Infrastructure and Security Lifecycle Management into code, automating workflows, and creating a central system of record based on a zero trust architecture. With infrastructure as code, platform teams can establish security guardrails and enable efficient provisioning at scale. This modern approach eliminates error-prone manual provisioning workflows and replaces them with standardized, secure modules and artifacts for reuse.

Organizations deploy The Infrastructure Cloud framework to manage their entire cloud estate's infrastructure and security lifecycle. Getting Devs and Sec teams on the same page starts by deploying secure cloud development tools that shift left to reduce team stress and enhance productivity. Organizations must prioritize software engineering alignment — improving cloud security rather than perpetuating siloed workflows with misaligned toolchains.

# Cheat sheet: 5 steps to realign your developer and security teams

You didn't cause this mess, but now you must fix your out-of-sync Dev and Sec teams. Is it possible to enhance cloud security and boost developer velocity? You held training sessions and threw team parties, but those efforts haven't stopped the friction. What's going to end the blame game? Here's how to sow "peace with platforms" and get everyone on the same page about cloud security.

## Engage your platform heroes

Platform teams typically choose the development tools and platforms that catalyze developer velocity. Equip them with tools that balance speed and security and eliminate long-standing tool and process misalignments between Dev and Sec teams.

## Start to shift left

Integrating security best practices earlier into developer workflows, known as a "shift-left" approach, satisfies both Devs and Sec because it fosters more secure software without slowing development.

## Terminate tool sprawl

Redundant tools are commonplace, but the penalty for them is not well understood. Beyond unnecessary licensing costs, organizations must contend with inconsistent security policies, potential vulnerabilities, and poor software integration.

## Embrace SLM

Security Lifecycle Management isn't a trendy lifestyle. It's a proven framework that enables developers to operate with agility while providing a centralized approach to controlling costs and protecting access to sensitive information.

## Automate workflows FTW

Deploy The Infrastructure Cloud, powered by the HashiCorp Cloud Platform (HCP), to replace error-prone manual provisioning workflows and standardize secure modules. Eliminate unnecessary tension and inefficiency and transform cloud security and application quality.

## About HashiCorp

HashiCorp is The Infrastructure Cloud™ Company, helping organizations automate multi-cloud and hybrid environments with Infrastructure Lifecycle Management (ILM) and Security Lifecycle Management (SLM). HashiCorp offers The Infrastructure Cloud on the HashiCorp Cloud Platform (HCP) for managed cloud services, as well as self-hosted enterprise offerings and community source-available products. The company is headquartered in San Francisco, California.

For more information visit hashicorp.com.