McKinsey Digital

Security as code: The best (and maybe only) path to securing cloud applications and systems

Managing security as code enables companies to create value in the cloud securely.

This article was a collaborative effort by Chhavi Adtani, Aaron Bawcom, Jan Shelly Brown, Rich Cracknell, Rich Isenberg, Kaz Kazmier, Pablo Prieto-Munoz, and David Weinstein, representing views from McKinsey Technology and McKinsey's Risk Practice.

© KrulUA/Getty Images

July 2021

Existing cybersecurity architectures and operating models break down as companies adopt public-cloud platforms. Why? Almost all breaches in the cloud stem from misconfiguration, rather than from attacks that compromise the underlying cloud infrastructure.[1]

So cloud requires secure configuration of applications and systems. But traditional cybersecurity mechanisms were not designed to ensure secure configuration or operate at the tempo required to capture the benefits of agility and speed that business leaders expect. As a result, as companies try to capture cloud value, they must adopt new security architectures and processes to protect their cloud workloads. Then cloud migration can increase not only the delivery of business value but also the security of their systems and applications compared with the old on-premises world.

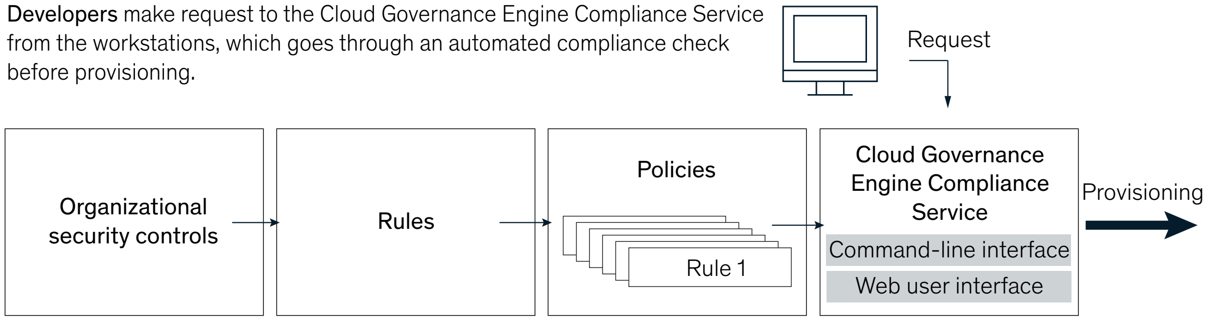
"Security as code" (SaC)[2] has been the most effective approach to securing cloud workloads with speed and agility. At this point, most cloud leaders agree that infrastructure as code (IaC) allows them to automate the building of systems in the cloud without error-prone manual configuration. SaC takes this one step further by defining cybersecurity policies and standards programmatically, so they can be referenced automatically in the configuration scripts used to provision cloud systems and systems running in the cloud can be compared with security policies to prevent "drift"[3] (Exhibit 1). If the business, for example, sets up a policy that all personally identifiable information (PII) must be encrypted when it's stored, that policy is translated into a process that is automatically launched whenever a developer submits code. Code that violates the PII policy is automatically rejected.

[1] Arul Elumalai, James Kaplan, Mike Newborn, and Roger Roberts, "Making a secure transition to the public cloud," January 2018, McKinsey.com.
[2] Patent information, US10872029B1: System, apparatus, and method for deploying infrastructure to the cloud.
[3] "Drift" refers to configuration modifications of an application from its original "as built" state, which occur as application owners change, update, or improve the product.

Exhibit 1

Security as code automates policy implementation.

Developers make request to the Cloud Governance Engine Compliance Service from the workstations, which goes through an automated compliance check before provisioning.



Organizational security controls capture list of security controls and operational policies that address control requirements.

Rules are small pieces of code that the Cloud Governance Engine Compliance Service checks IaC against to ensure the requested resources to be provisioned will comply with organization compliance requirements.

Policies group rules together to define compliance requirements for a given context and are organized hierarchically, defining a compliance inheritance structure.

Cloud Governance Engine Compliance Service is the centralized service responsible for accepting provisioning requests and verifying IaC meets organization compliance requirements prior to provisioning cloud resources.

Capturing value in the cloud requires most companies to build a transformation engine (Exhibit 2) to integrate cloud into business and technologies, drive adoption in priority business domains, and establish the foundational capabilities required to scale cloud usage safely and economically. SaC is the mechanism for developing foundational capabilities in cloud security and risk management.

Yes, SaC could theoretically be implemented in a company's own data centers, but the automation capabilities available from cloud service providers (CSPs) make it much more practical to do it in the cloud.

The three most powerful benefits of SaC

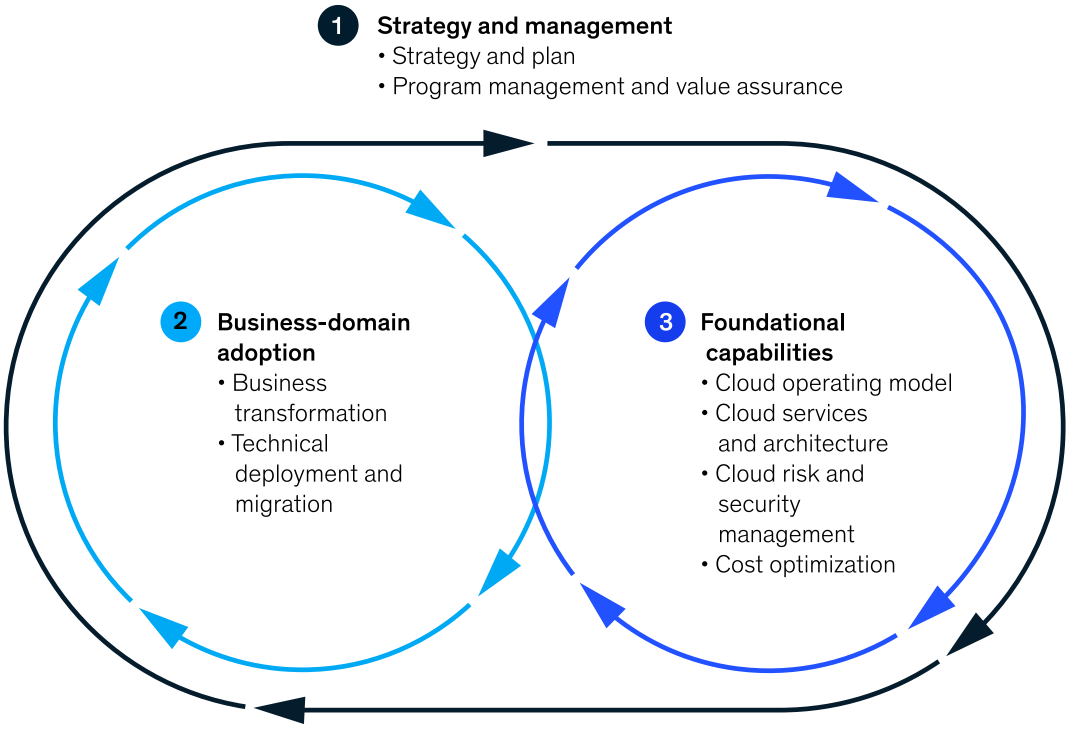
The first benefit of SaC is speed. To fully realize the business benefits of the cloud, security teams must move at a pace they are unaccustomed to in on-premises environments. Manual intervention introduces friction that slows down development and erodes the cloud's overall value proposition to the business.

The second benefit is risk reduction. On-premises security controls simply don't account for the nuances of the cloud. Cloud security requires controls to move with a workload throughout its entire life cycle. The only way to achieve this level of embedded security is through SaC.

Finally, SaC is a business enabler. Security and compliance requirements are becoming increasingly central to businesses' core products and services. In this respect, SaC not only expedites time to market but expands opportunities for innovation and product creativity without compromising security.

Exhibit 2

The cloud transformation engine is made up of three mutually reinforcing elements.



1 **Strategy and management**
- Strategy and plan
- Program management and value assurance

2 **Business-domain adoption**
- Business transformation
- Technical deployment and migration

3 **Foundational capabilities**
- Cloud operating model
- Cloud services and architecture
- Cloud risk and security management
- Cost optimization

Implementing the SaC approach

Implementing SaC requires a substantial policy, architectural, and cultural departure for almost all companies. For this reason, many have found it helpful to use the SaC framework to classify workloads according to sensitivity and criticality and then apply specific controls, based on workload risk and deployment type. Furthermore, it provides a blueprint for a future-state architecture and outlines the key decisions to be made in order to effectively and efficiently instantiate SaC through various automation use cases. Finally, the framework defines how the company's operating model must change to extract the full benefits of cloud adoption (Exhibit 3).

Risk classification, deployment model, and standards/policies

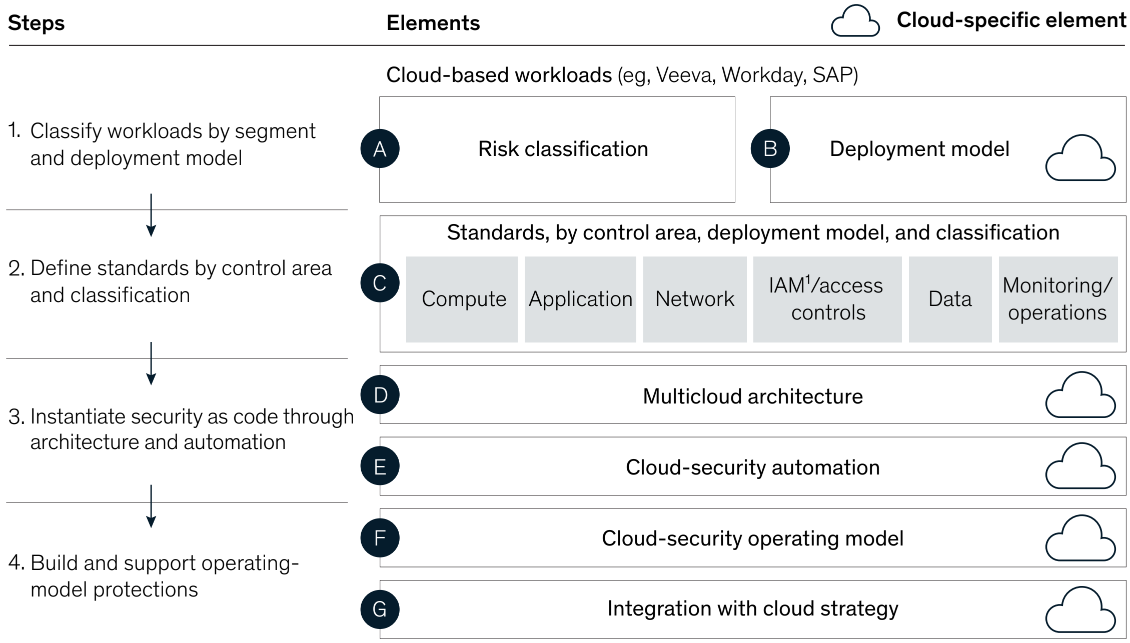
After classifying the sensitivity and criticality of the workload and its data, the next step is to apply specific policies that can ultimately be instantiated as code. To get to this level, organizations typically encounter three choices for each control area.

The first is to identify and write new policies to account for technologies that are not present on-premises. Consider container security, for example. Companies that have historically operated on-premises are likely not enforcing the scanning of containers for vulnerabilities at ship, deploy, and runtime. Similarly, there should be policies in place for using trusted images and secure registries. Another example of a common gap in standards is API security and the need to authenticate traffic and control the use of APIs with a gateway.

In addition to writing new policies, companies should modify existing policies for on-premises environments to address the unique security context of the cloud. Consider network segmentation, for example. Whereas most on-premises organizations enforce network-segmentation policies limited to firewall rules at coarse-grained network boundaries, in the cloud they can configure security groups to provide network protections down to the host level, effectively providing micro-segmentation even within a single subnet. In addition, they can still use

Exhibit 3

Security as code can be implemented through four steps.


Steps
Elements
Cloud-specific element
Cloud-based workloads (eg, Veeva, Workday, SAP)
1. Classify workloads by segment and deployment model
A
Risk classification
B
Deployment model
2. Define standards by control area and classification
C
Standards, by control area, deployment model, and classification
Compute
Application
Network
IAM[1]/access controls
Data
Monitoring/ operations
3. Instantiate security as code through architecture and automation
D
Multicloud architecture
E
Cloud-security automation
4. Build and support operating-model protections
F
Cloud-security operating model
G
Integration with cloud strategy

[1] Identity and access management.

the more coarse-grained controls at the subnet or virtual-network perimeter level. These can all be configured and maintained by code rather than manual processes and runbooks.

Finally, to fully realize the promise of SaC, all polices—existing and new—must be written to a level of detail sufficient to enable them to be translated into code. One of the biggest gaps we observe in the making of effective security policy is the scarcity of people who understand how to design effective policy and policy implementation. Poorly designed policy and policy implementation can be as problematic and costly as the risks they're intended to address. Perhaps the most common example relates to application security. Policies must be granular enough to automatically prevent applications from going into production unless they have no outstanding vulnerabilities that exceed a prescribed remediation timeframe. Similarly, for identity-and-access-management purposes, policies should require the integration of software-as-a-service (SaaS) applications with a federated-identity system to automatically enforce multifactor authentication (MFA) based on the application's risk classification.

Architecture and automation

Once the organization has established a robust risk-classification framework and defined its policies for cloud, successful implementation of SaC depends on making key architectural-design decisions and executing the right automation capabilities. As with policies, we recommend mapping these decisions to the organization's priority control areas.

But before a company can answer key questions on its cloud security architecture, it must first make design decisions about its overall cloud architecture. This seems obvious, but it is a critical step that is often overlooked. The organization can begin to design the security architecture only after it has answered key questions about its multicloud architecture—including (but not limited to) the selection of the primary cloud provider and how the strategic perimeter will be designed.

Perhaps the most important decision for implementing SaC is whether policy objectives will be instantiated via federated, native CSP tools or centralized, third-party tools. There is no inherently right or wrong answer; rather, this question must be considered in the context of the organization's risk appetite, workload classification, policy objectives, and overall cloud strategy. A rationale for centralized, third-party tooling is to enable integrated visibility across on-premises and cloud environments through a single pane of glass.

Automation is fundamental to enforcing policies at different phases of the systems development life cycle (SDLC). It's important, therefore, for organizations to develop a comprehensive list of security-automation use cases for their multicloud environment. Unlike the policy objectives and architecture capabilities that were mapped to control areas, these use cases should map to the phases of the SDLC: code, build, package, deploy, and operate. Some use cases, such as system decommissioning and failover testing, are unique to infrastructure-as-a-service (IaaS) deployments, while others, such as log transmission, ingestion, and code scanning, are common across multiple deployment models. A Cloud Governance Engine Compliance Service is used to manage the process. It is the centralized service responsible for accepting provisioning requests and verifying IaC meets organization compliance requirements prior to provisioning cloud resources.

Operating model and linkage to broader cloud strategy

When embarking on a journey to the cloud and adapting their security accordingly, most companies take a technology-first approach. As important as technological change is, however, companies need to evolve their operating model to "shift left," maximize self-service, and achieve full-life-cycle security automation.

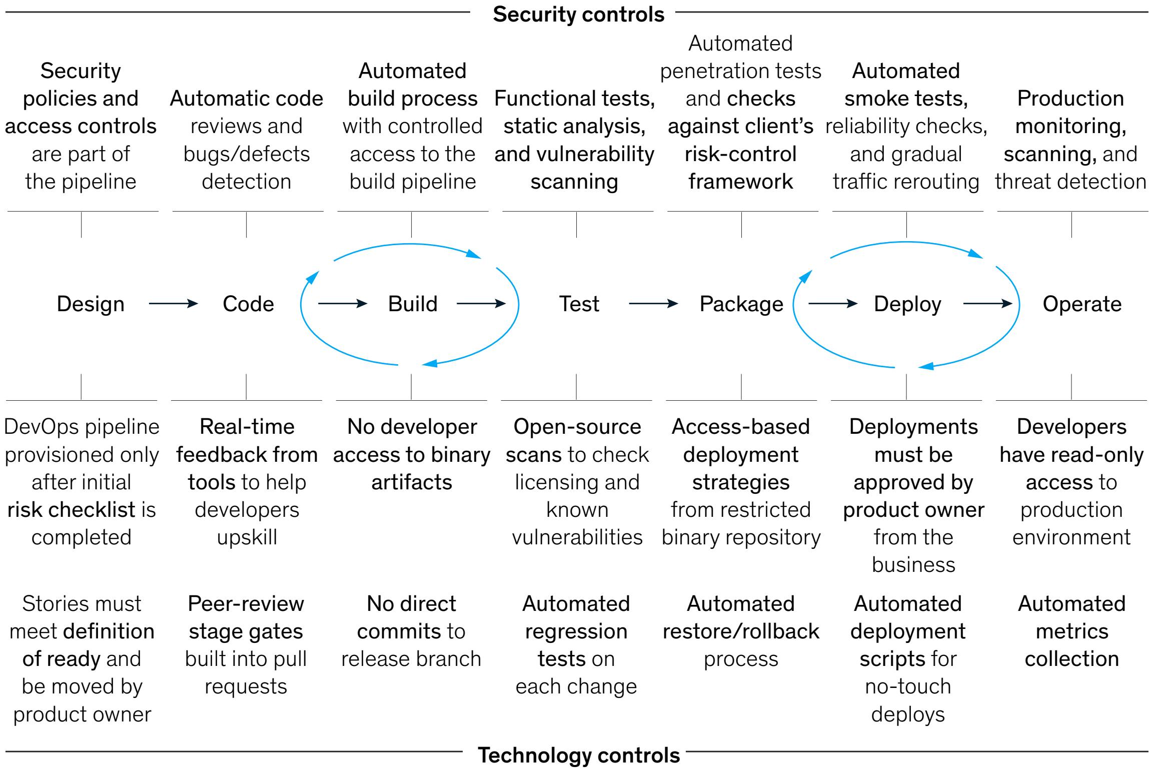
SaC requires highly automated services that developers can consume via APIs. This, in turn, requires behavioral changes in security, infrastructure, and application-development teams. The security organization must move from a reactive, request-based model to one in which they engineer highly automated security products—for example, in identity and access management or vulnerability management. In the cloud, many controls are simply configuration choices for core compute or storage services. As a result, infrastructure-product engineering teams have to collaborate much more closely with their security counterparts to apply security requirements to their product backlogs. Development teams require site-reliability engineers (SREs) sophisticated enough to consume security services effectively in provisioning and managing applications.

To achieve this, companies need to align a common development toolchain and continuous-integration and continuous-delivery (CI/CD) pipelines (Exhibit 4). Unless these capabilities are standardized, SaC—and therefore security in the cloud—becomes all but impossible. They must also upgrade skills across the technology organization. SREs need to be sophisticated security consumers. Infrastructure-product managers and engineers need to understand the security implications of their products. And the security organization needs more product-management and engineering capabilities than compliance.

Exhibit 4

Introducing control automation in the CI/CD pipeline both mitigates risk and increases speed of delivery.

Illustrative



	Design	Code	Build	Test	Package	Deploy	Operate
Security controls	Security policies and access controls are part of the pipeline	Automatic code reviews and bugs/defects detection	Automated build process with controlled access to the build pipeline	Functional tests, static analysis, and vulnerability scanning	Automated penetration tests and checks against client's risk-control framework	Automated smoke tests, reliability checks, and gradual traffic rerouting	Production monitoring, scanning, and threat detection
Technology controls	DevOps pipeline provisioned only after initial risk checklist is completed	Real-time feedback from tools to help developers upskill	No developer access to binary artifacts	Open-source scans to check licensing and known vulnerabilities	Access-based deployment strategies from restricted binary repository	Deployments must be approved by product owner from the business	Developers have read-only access to production environment
Technology controls	Stories must meet definition of ready and be moved by product owner	Peer-review stage gates built into pull requests	No direct commits to release branch	Automated regression tests on each change	Automated restore/rollback process	Automated deployment scripts for no-touch deploys	Automated metrics collection

Views on SaC from CISOs

CISOs from Amazon, Google, and Microsoft see multiple benefits in security as code.

Mitigating human error:

"Security through code is all about repeatability. We implement automation and use of code for security purposes because it applies universal rigor throughout the organization. It solves the issue of human error that is the common denominator across cloud breaches."

—Stephen Schmidt, AWS CISO

"Over time, economies of scale will drive higher-level controls by default. The more we can raise the baseline of controls (for example, encrypted virtual machines, high-assurance container workloads, identity and access management, customer-managed encryption keys), the more customers can unlock all classes of workloads. Shortly, we'll reach a tipping point when such enhanced security controls will be deployed across all services for all customers."

—Phil Venables, Google Cloud CISO

Balancing speed and risk:

"Security as code is all about giving our folks the freedom to do work while ensuring we have visibility into what they're doing so there aren't hidden behaviors or shadow IT. The security team must play an auditor role in accounts to see with specificity what is going on and get notified when changes are made."

—Stephen Schmidt, AWS CISO

"The automation capabilities and insights you gain across your inventory, services, configurations, and the ability to patch at scale are just not possible on-premises. We were also able to reduce our supplier footprint by 48 percent over two and a half years by leveraging integrated, holistic cloud solutions."

—Bret Arsenault, Microsoft CISO

"The pace of security enhancement and extent of security-feature additions to our products (our theme of secure products, not just security products) are accelerating. Many other cloud providers have made similar progress. This massive, global-scale competition to keep increasing security in tandem with agility and productivity is a benefit to all."

—Phil Venables, Google Cloud CISO

Lifting and shifting security:

"The biggest problem we see is organizations trying to forklift security from on-premises to the cloud. Doing so misses so many opportunities that exist in cloud but not the data center."

—Stephen Schmidt, AWS CISO

"For some customers without sophisticated security teams, all they have to do is take every new security feature the cloud offers and keep them enabled. The most advanced customers requesting new features can drive benefit for all, and this—along with the visibility into and responsiveness to threats that we have as a hyperscale cloud provider—creates, in effect, a digital immune system for the whole customer base."

—Phil Venables, Google Cloud CISO

"We can relate to enterprise customers; our security infrastructure and tools were not born in the cloud either. However, organizations cannot just take traditional, on-premises security processes and tools and apply them in a cloud environment. They simply won't function at the necessary scale or speed. They must be transitioned for the environment they are operating in. It's an investment, but the benefits, including cost savings overall, are significant."

—Bret Arsenault, Microsoft CISO

Automating containment:

"Some organizations can automatically remediate down to known good state, but most of the world is not there [and] should focus on automated containment instead. All components of an application need to have specific, up-to-date, and tested runbooks for containment in event of unexpected behavior. Bring in the business and application owners early. You don't want to debate about when to pull the cord that isolates everything in the heat of battle."

—Stephen Schmidt, AWS CISO

How one company implemented SaC

One wholesale financial institution was looking to move 80 percent of its workloads to cloud. It quickly became apparent, however, that the traditional approach of reactively addressing compliance issues after they had been deployed to production environments was unable to keep up with the cloud workloads. So the company decided to proactively drive compliance in the pipeline through SaC.

To make this happen, it classified about 400 controls for its mission-critical applications to ensure a resilient cloud infrastructure. It then developed and implemented more than 90 rules to support those controls and translated them into small pieces of code that are invoked anytime any infrastructure provisioning requests are made. These blocks of code ensure that all compliance requirements are met prior to provisioning. This approach currently provides automated coverage for about half of all security requirements.

Too often, security is viewed as an obstacle to cloud adoption. What should be a frictionless deployment process with security embedded at the outset becomes weeks or months of back-and-forth between developers, infrastructure, and security as they try to shoehorn cloud deployments into legacy security mechanisms. Lengthy approvals ranging from third-party assessments to firewall changes not only decrease the overall value proposition of the cloud but also increase the need for risky policy exceptions to accommodate business requirements.

SaC endeavors to flip this trend on its head and position CISOs as enablers, not impediments, to the business. Eliminating the need for ticket-based workflows, escalations, and manual change processes, a SaC framework automates the most critical security controls and enhances overall security compared with on-premises systems—without slowing down the business or sacrificing compliance needs.

Chhavi Adtani is an associate partner in McKinsey's Chicago office; **Aaron Bawcom** is a delivery senior expert in the Atlanta office, where **Rich Isenberg** is a partner; **Jan Shelly Brown** and **David Weinstein** are associate partners in the New Jersey office; **Rich Cracknell** is a solution leader in the Silicon Valley office; **Kaz Kazmier** is a delivery expert in the Seattle office; and **Pablo Prieto-Munoz** is an associate partner in the New York office.