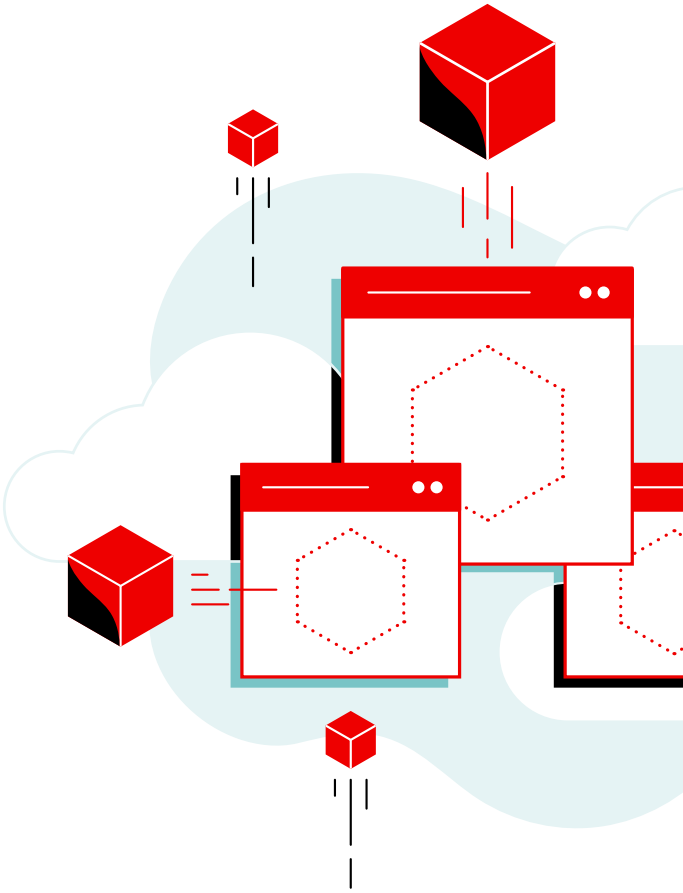
Red Hat

State of Kubernetes Security Report



Spring 2021

Executive Summary

This edition of the State of Kubernetes Security Report examines how companies are adopting Kubernetes, containers, and cloud-native technologies while meeting the challenges of securing their vital Kubernetes applications. This report compiles the survey results from more than 500 DevOps, engineering, and security professionals and uncovers new findings about how companies that embrace containers and Kubernetes implement DevSecOps initiatives to protect their cloud-native environments. The survey was conducted by StackRox before its acquisition by Red Hat in early 2021.

Because security as the biggest area of concern with container adoption and security issues continue to cause delays in deploying applications into production, we also look at the most common types of security incidents that companies experience in their Kubernetes environments.

The survey results highlight the importance of collaboration across Dev, Ops, and Security teams to implement security early in the development life cycle to realize the greatest benefit of Kubernetes—innovating fast. We are heartened to see so many organizations adopting DevSecOps—75% of organizations have initiatives in place that increase collaboration between DevOps and Security teams.

Nearly everyone—94% of respondents—admitted to experiencing a security incident in the last 12 months. In many cases, the cause was a misconfiguration. But a sizable portion also identified a major vulnerability, experienced a runtime incident, or failed an audit. These findings become more critical when respondents have deployed their Kubernetes workloads in production environments.

We encourage you to benchmark yourself against the findings in this report to determine how you can accelerate your efforts to apply security controls across containers and Kubernetes. Delaying security could mean delaying innovation and putting the business benefits of Kubernetes at risk. There are many security advantages you can use in containers and Kubernetes—from declarative configuration and immutable infrastructure to the isolation inherent in containerized applications. Organizations, however, need the knowledge, tooling, and processes to put those capabilities to work so they can benefit from the sizable advantages of running fast in a DevOps-driven, cloud-native world.

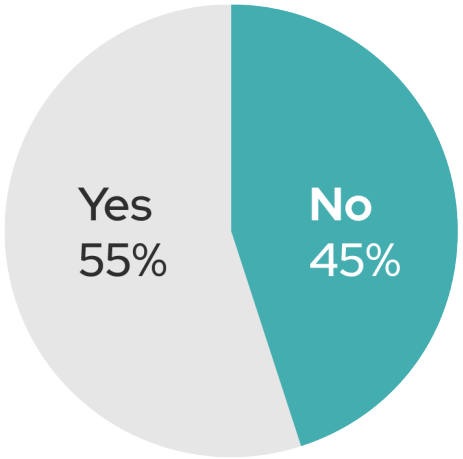
Security concerns are slowing down innovation

More than half of respondents have delayed deploying Kubernetes applications into production due to security

Companies are rapidly adopting Kubernetes and containers to fuel the growth engine for their digital innovation and transformation. A slew of new companies—born in the cloud—rely on this tech stack to fuel their growth, while incumbents are migrating their existing workloads to containers and Kubernetes across hybrid cloud environments.

Rapid application development and release, swift bug fixes, and increased feature velocity are three of the most often cited benefits of containerization. However, when security becomes an afterthought, you might end up negating the greatest gain of containerization—agility. More than half of the survey respondents (55%) have had to delay an application rollout because of security concerns. Rolling out an application that hasn't passed a security assessment puts the business at significant risk. To prevent delays in application deployment and realize the benefits of containers and Kubernetes, organizations must shift left with security, building it into the development phase so they can address as many security challenges as possible during the build stage.

Have you ever delayed or slowed down application deployment into production due to container or Kubernetes security concerns?



94% of respondents experienced at least one security incident in their Kubernetes environments in the last 12 months

Misconfiguration is the leading cause of security incidents, by a wide margin

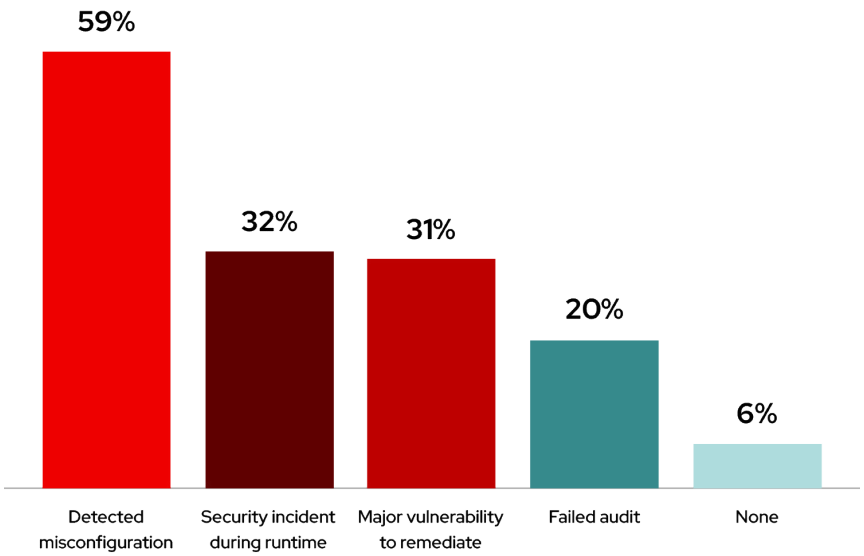
A whopping 94% of survey respondents have experienced a security incident in their Kubernetes and container environments during the last 12 months.

The fact that nearly everyone has had a security problem may help explain the previous finding, that over half of organizations have delayed an application deployment over security concerns.

Human error is the most often cited cause of data breaches and hacks. Kubernetes and containers, while powerful, increase this risk substantially. Kubernetes has powerful functionality applied through a declarative model. A single workload may require significant configuration to ensure a more secure and scalable application. Add on technical debt and organizational hurdles, and it is a challenge even for experienced Kubernetes professionals to get everything right all the time.

Not surprisingly, nearly 60% of respondents have experienced a misconfiguration incident in their environments over the last 12 months. Nearly a third have discovered a major vulnerability, and another third said they've suffered a runtime security incident. Lastly, 20% said they had failed an audit.

In the past 12 months, what security incidents or issues related to containers and/or Kubernetes have you experienced?



Security tops the list of concerns with container strategies

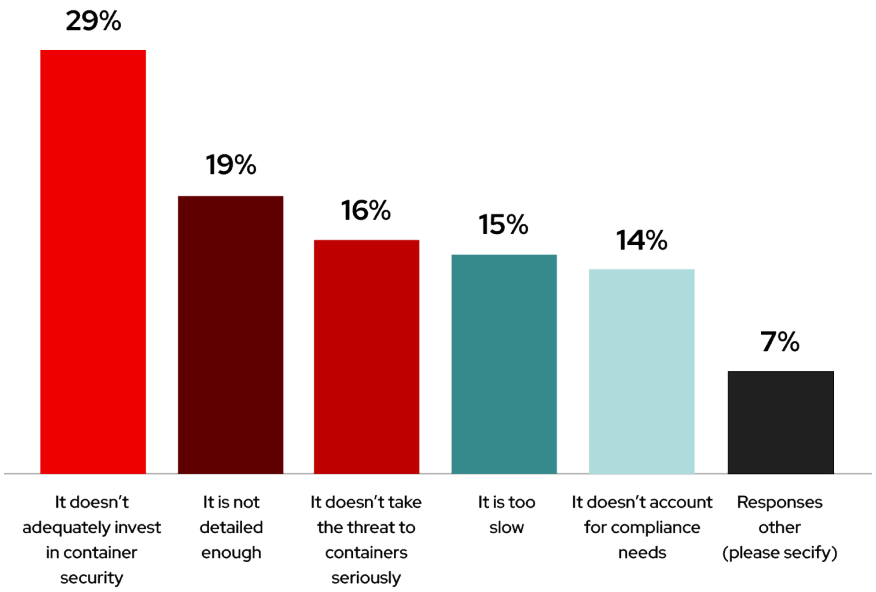
59% of respondents are most worried about unaddressed security and compliance needs or threats to containers

Given what we know about the prevalence of security incidents in these environments (94%), it should come as no surprise that security remains the top concern when it comes to container adoption.

Inadequate investment in security is the top-cited concern about the respondent company's container strategy (29%). When combined with not taking threats seriously (16%) and not accounting for compliance needs (14%), nearly two-thirds of respondents identify security and compliance as their biggest source of concern.

Organizations are eagerly adopting containers and Kubernetes. If they don't make the necessary investments in security strategies and tooling simultaneously, they risk the security of their critical applications and may need to delay application rollout.

What is your biggest concern about your company's container strategy?



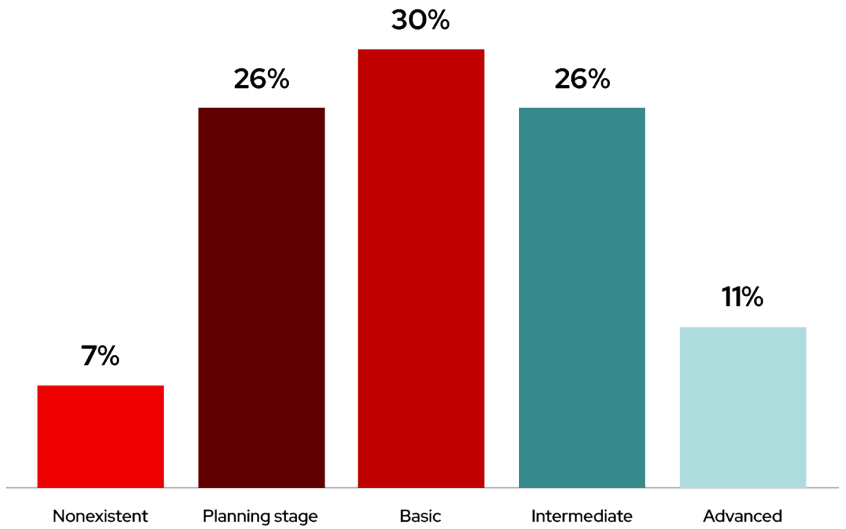
Majority has moved past the planning stage of their container security strategies

More than a third have an intermediate or advanced security strategy

With growing container adoption, organizations continue to advance their container and Kubernetes security strategies. The percentage of respondents with at least a basic Kubernetes security strategy is at 67%. Even more notable is the percentage of respondents who lack a security strategy entirely; that number is just 7%.

While this data is promising, the previous finding—quantifying the ongoing security concerns about container strategies—shows that while security strategies are maturing, organizations still need to make further investments in their plans so they can adequately address container security and compliance needs.

How would you describe the security strategy for your company's container and Kubernetes environments?



Responsibility for Kubernetes security is highly decentralized

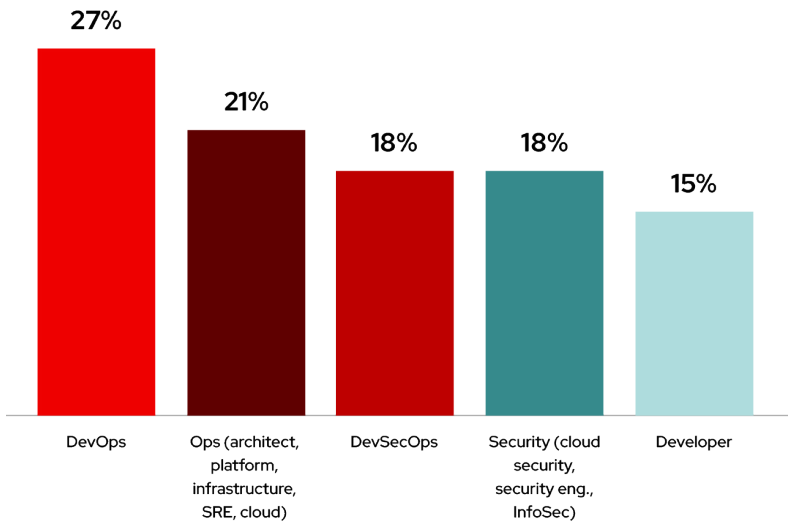
Across various roles, DevOps is considered most responsible for security

Across various roles, DevOps is the single role most cited as responsible for securing containers and Kubernetes. Taken together, the myriad operational roles of DevOps, Ops, and DevSecOps are considered the primary owners of Kubernetes security by a whopping 66% of respondents.

Echoing the need for security to shift left, 15% of respondents consider developers as the primary owners of Kubernetes security, with only 18% identifying security teams as being most responsible.

This distribution shows that when it comes to container and Kubernetes security, it takes a village. Traditionally, Security has been the central control point for enforcing security and compliance policies. Containers and Kubernetes adoption are often primarily driven by DevOps, so it's not surprising to see respondents naming them responsible for securing these technologies. To bridge these gaps, container and Kubernetes security tooling must facilitate close collaboration among different teams—from Developers to DevOps to Ops to Security—instead of perpetuating the barriers that may plague organizations.

What role at your organization is most responsible for container and Kubernetes security?



Most organizations have a DevSecOps initiative

Only 26% of respondents say DevOps and Security remain separate

DevSecOps is no longer just a buzzword—the term encompasses the processes and tooling that allows security to be built into the application development life cycle, rather than as an afterthought. Our survey found good news on this front—the vast majority of respondents say they have some form of DevSecOps initiative underway. Only 26% of respondents continue to operate DevOps separate from Security.

Even more promising is that 25% of respondents have an advanced DevSecOps initiative, where they're integrating and automating security throughout the life cycle.

Do you have a DevSecOps initiative in your organization?

Misconfigurations pose the greatest security concern for respondents

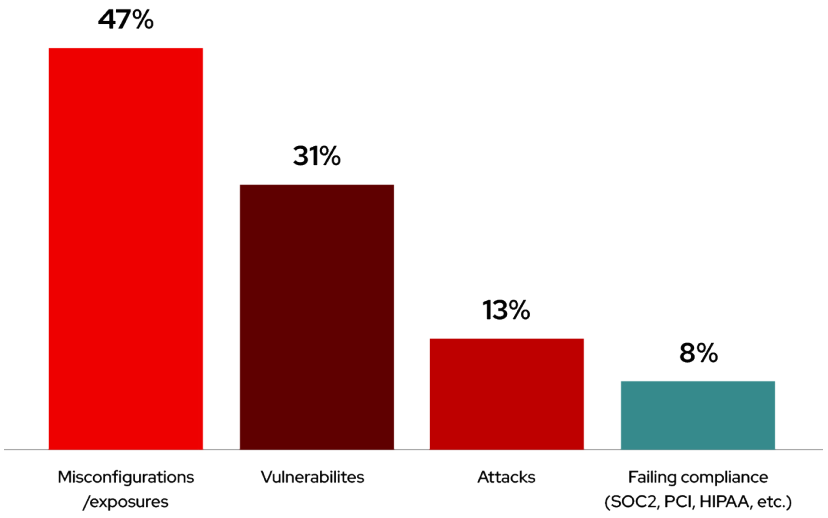
Prevalence of misconfigurations underscores the need for automating configuration management

Survey respondents worry the most about exposures due to misconfigurations in their container and Kubernetes environments (47%)—almost four times the level of concern over attacks (13%), with vulnerabilities as the second leading cause of worry (31%).

Configuration management poses a uniquely difficult challenge for security practitioners. While a host of tools are available for vulnerability scanning of container images, configuration management requires more consideration. People may know that they should avoid deploying the Kubernetes dashboard, but configuring a pod's security context or implementing Kubernetes role-based access control (RBAC) are just two examples of more challenging settings that teams need to get right.

The best way to address this challenge is to automate configuration management as much as possible, so that security tools—rather than humans—provide the guardrails that help developers and DevOps teams configure containers and Kubernetes securely.

Of the following risks, which one are you most worried about for your container and Kubernetes environments?



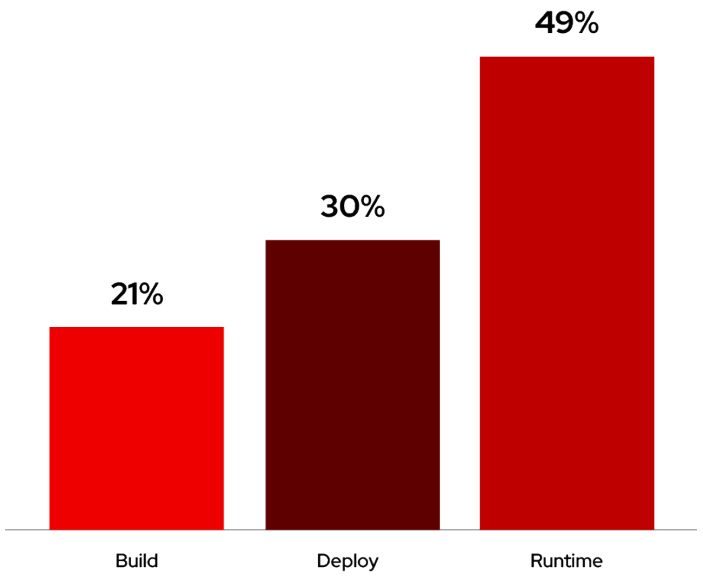
Respondents worry the most about the runtime phase of the container life cycle

Nearly half of respondents are most worried about the runtime phase, underscoring the importance of runtime controls

Runtime is the container life-cycle phase that organizations worry about the most. At first glance, this finding is counter to our previous finding, given that an overwhelming majority of respondents identify misconfigurations as the source of biggest security risk, and have experienced a misconfiguration incident more often than any other type of security incident.

However, the data makes more sense when you consider that runtime security issues are usually caused by security lapses, such as a misconfiguration, at build or deploy stage. Furthermore, any negative outcome of a security misstep at build or deploy stages is likely to be felt only once an application is running in production.

Which life-cycle phase are you more worried about from a security perspective?



Hybrid cloud deployment strategies are the most common

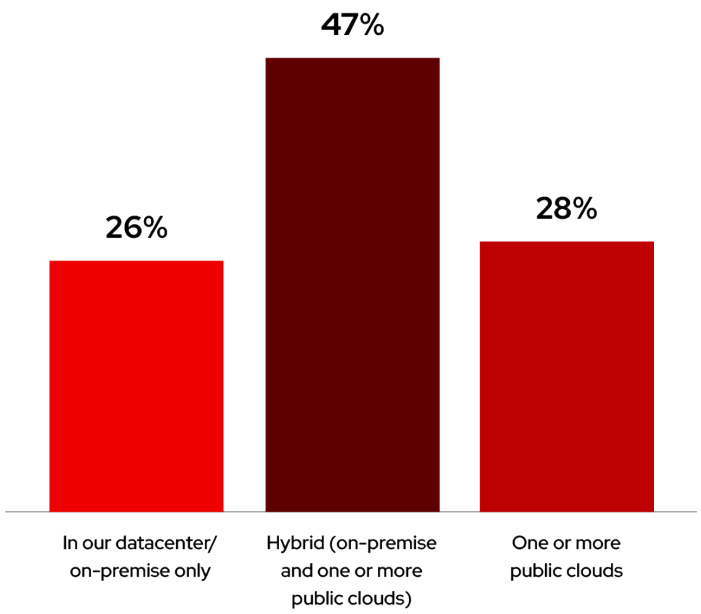
47% of respondents deploy containers across a hybrid cloud model while 28% have selected a cloud-only strategy

Talk of cloud-only strategies runs high, but actual deployments on one or multiple cloud providers only still lags hybrid cloud deployments—47% of respondents run their containers in a hybrid setting vs. 28% who run only in public cloud.

When isolating on-premise-only container deployments, the delta still remains, with 26% running containerized applications in their own data centers only.

With hybrid models continuing to be the dominant approach, organizations need security that runs the same way no matter where workloads are deployed. Security approaches that are Kubernetes-native can deliver environment-agnostic controls that span on-premise to cloud deployments.

Where do you have containers running?



Red Hat OpenShift is the leader in hybrid cloud deployments

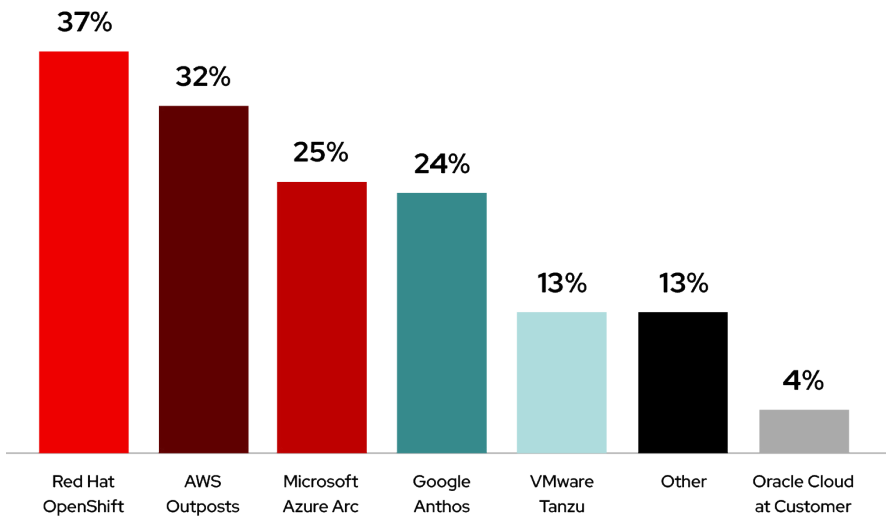
37% of respondents have standardized on OpenShift, with AWS Outposts and Azure Arc rounding out the top three

With hybrid cloud deployments the most popular mode of running containerized applications, we wanted to understand how organizations were deploying in hybrid mode.

The popularity of technologies from the public cloud providers follows a similar arc of overall platform popularity. However, they all lag behind Red Hat® OpenShift®, according to our survey.

The hybrid offerings from VMware and Oracle lag behind their peers, with 13% and 4% of respondents using them respectively.

Are you using any solutions for hybrid and multicloud Kubernetes deployments?



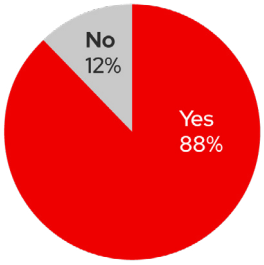
As the de facto container orchestrator, Kubernetes is used by nearly everyone

88% of respondents use Kubernetes as their container orchestrator, with 74% in production

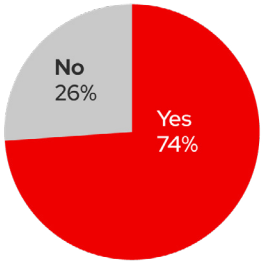
Our survey results indicate widespread customer adoption of Kubernetes (87%), especially in production environments (75%). But how they're deploying Kubernetes continues to change dramatically.

Amazon's EKS is the most widely used Kubernetes platform according to respondents (51%). Self-managed Kubernetes—somewhat surprisingly—is the second most commonly selected platform (35%), with Red Hat OpenShift coming in at a close third (33%). Microsoft's AKS and Google's GKE are neck and neck, with Azure holding a slight advantage.

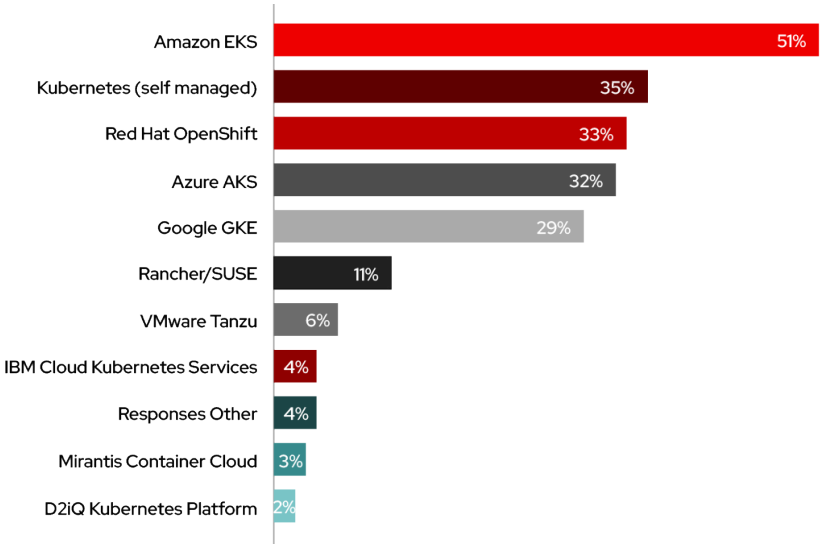
Are you using Kubernetes for container orchestration?



If you're using Kubernetes, are you running workloads in production?



What Kubernetes platform do you use to orchestrate your containers?



Respondents require a feature-rich security solution

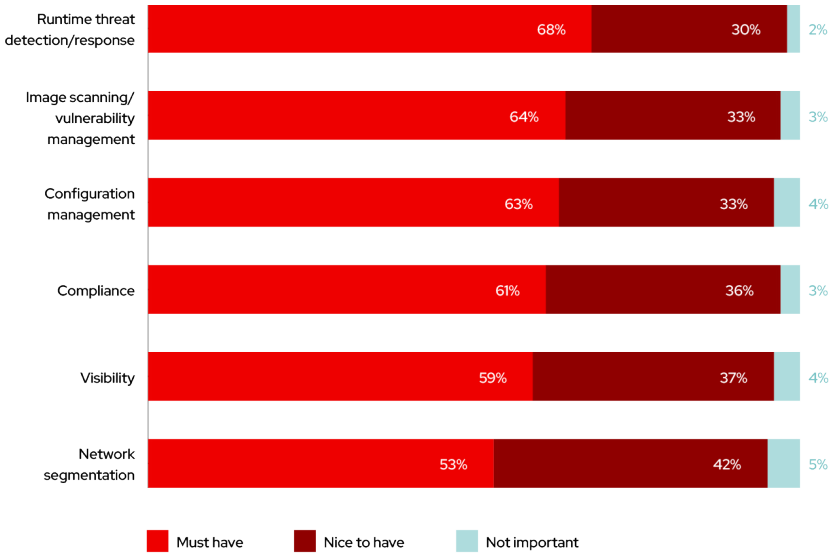
68% of respondents identify runtime threat detection and incident response as a must-have capability

Respondents expect a lot out of their Kubernetes security platforms—with two-thirds of the respondents citing all but network segmentation as a must-have capability. The capabilities span DevOps and security activities, underscoring the need for both broad and deep functionality in container and Kubernetes security platforms.

This breadth also highlights the fact that securing Kubernetes and containers requires involvement from Dev, Ops, and Security teams.

Respondents identified image scanning/vulnerability management (64%) and configuration management (64%) as two of the top three security capabilities they consider as must-have. At least half of the respondents identified each of these security capabilities as must-have.

How would you rate the importance of the following Kubernetes security capabilities?



Respondents use a wide variety of open source security tools

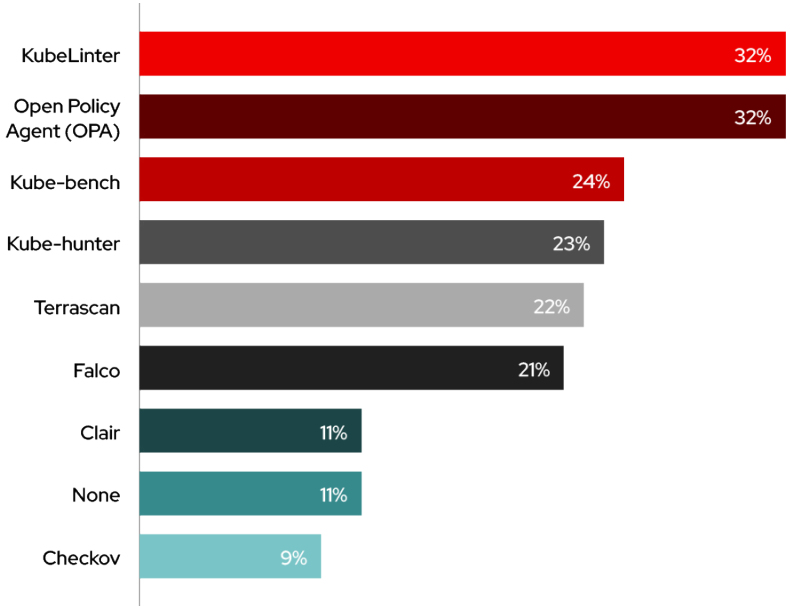
Six different open source security tools are used by at least 20% of respondents, with KubeLinter and OPA as the top two

Kubernetes is, first and foremost, a tool for development and DevOps teams to accelerate and scale containerized application development, deployment, and management. Providers such as Red Hat, Amazon, Microsoft, and Google have added security features to enhance the base capabilities in Kubernetes. At the same time, commercial security vendors have stepped up to offer enterprise-ready security solutions for more advanced use cases.

In parallel, the Kubernetes community has been very active in releasing open source security tools to fill in the security gaps present in Kubernetes. Customers have a rich selection of open source security tools to choose from, and our survey results shows that no single open source security tool dominates the Kubernetes security market.

KubeLinter and OPA are two of the most popular OSS for security, but the difference between the third place Kube-bench (24%) and sixth place Falco (21%) is only three percentage points.

What open source tools do you use for Kubernetes security?



Key takeaways for your container and Kubernetes security journey

The findings in this survey of over 500 respondents highlight the fact that organizations are putting at risk the core benefit of faster application development and release by not ensuring their cloud-native environments are built, deployed, and managed securely. With the prevalence of misconfigurations and vulnerabilities across organizations, security must shift left to be embedded into DevOps workflows instead of "bolted on" when the application is about to be deployed into production. With over half of our respondents delaying production deployment because of security concerns, a lack of security controls could inhibit business acceleration and innovation.

1. Use Kubernetes-native security architectures and controls.

Kubernetes-native security uses the rich declarative data and native controls in Kubernetes to deliver several key security benefits. Analyzing the declarative data available in Kubernetes yields better security, with risk-based insights into configuration management, compliance, segmentation, and Kubernetes-specific vulnerabilities. Using the same infrastructure and its controls for application development and security reduces the learning curve and enables faster analysis and troubleshooting. It also eliminates operational conflict by ensuring security gains the same automation and scalability advantages that Kubernetes extends to infrastructure.

2. Implement full life-cycle security, from build/deploy to runtime.

Security has long been viewed as a business inhibitor, especially by developers and DevOps teams whose core mandates are to deliver code fast. With containers and Kubernetes, security should become a business accelerator by helping developers build strong security into their assets right from the start. Look for a container and Kubernetes security platform that incorporates DevOps best practices and internal controls as part of its configuration checks. It should also assess the configuration of Kubernetes itself for its security posture, so developers can focus on feature delivery.

3. Require portability across Kubernetes environments.

With most organizations deploying containers in both on-premise and public cloud environments (sometimes in multiple clouds), security must apply consistently wherever your assets are running. The common foundation is Kubernetes, so make Kubernetes your source of truth, your point of enforcement, and your universal visibility layer so you have consistent security. Managed Kubernetes services may quicken your organization's ability to adopt Kubernetes, but be careful about getting locked into cloud provider-specific tooling and services.

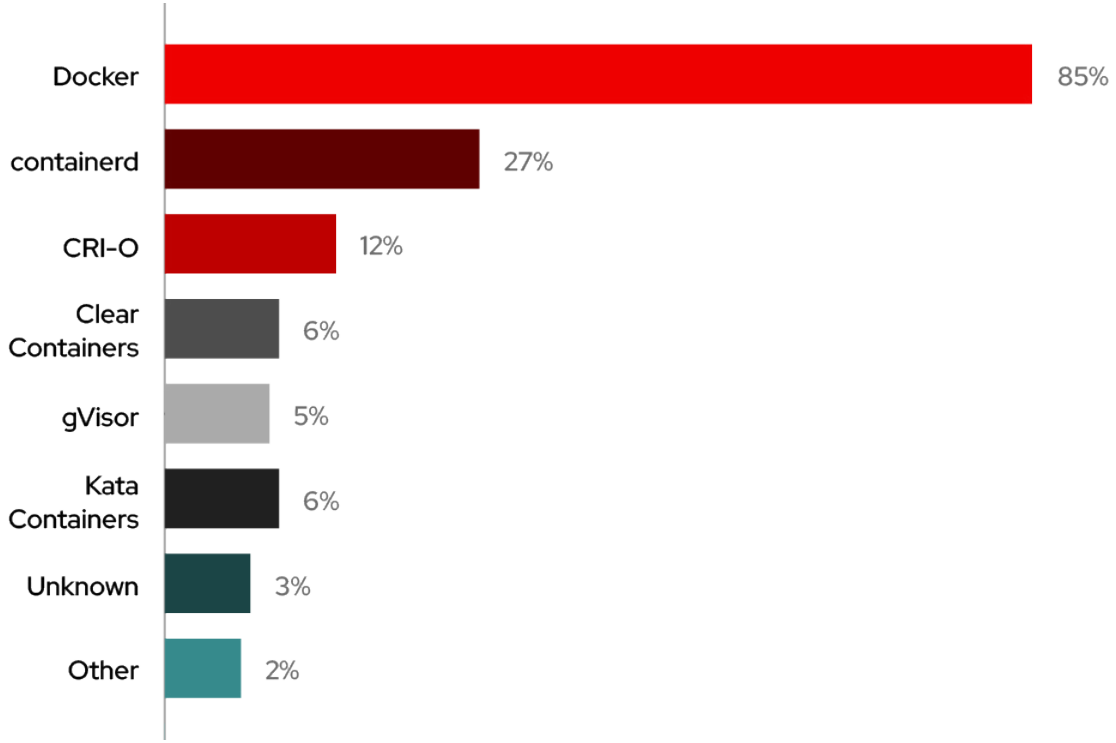
4. Build a bridge between DevOps and Security.

Given most organizations expect either DevOps or Security teams to run container security platforms, your security tooling must help bridge Security and DevOps. To be effective, the platform must have security controls that make sense in a containerized, Kubernetes-based environment. It should also assess risk appropriately. Telling a developer to fix all 39 discovered vulnerabilities with a CVSS score of 7 or higher is inefficient. Identifying for that developer the three deployments that are exposed to that vulnerability, and showing why they're risky, will get you action that will genuinely improve your security posture.

About our respondents—container runtime technology

Docker runtime engine remains dominant, with containerd a distant second

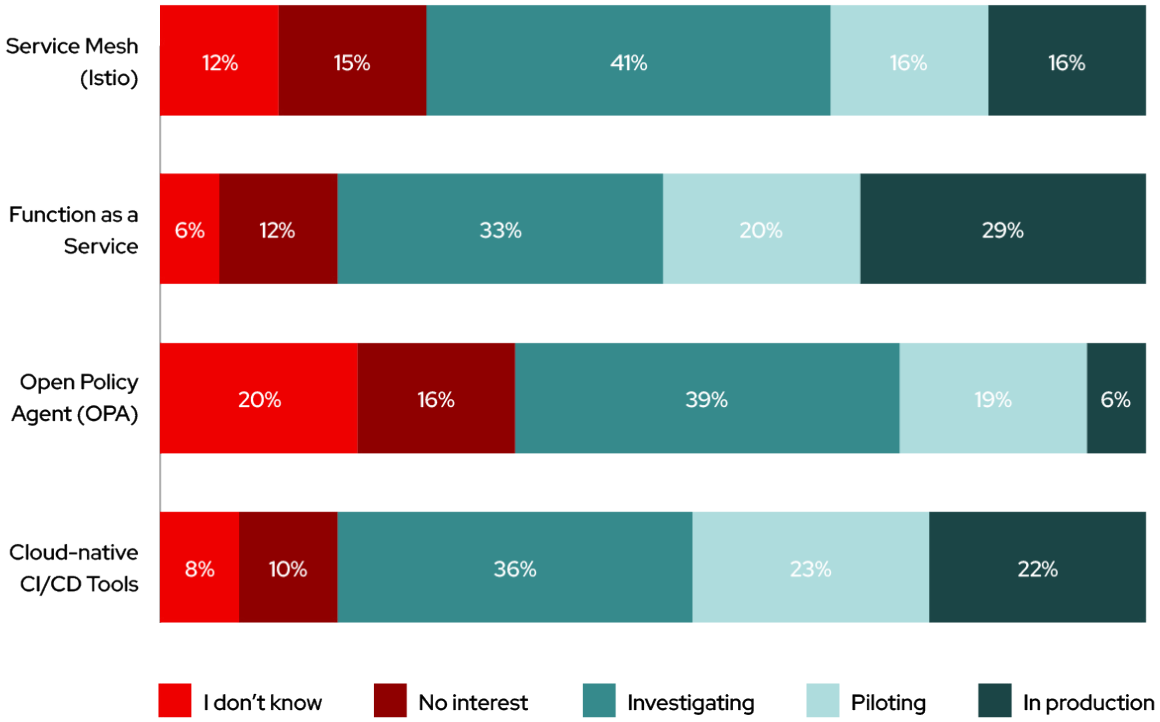
What container runtime(s) do you use?



About our respondents—other cloud-native technologies

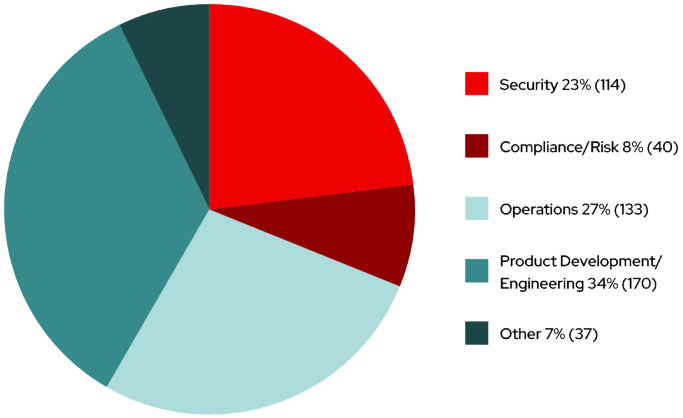
Emerging cloud-native technologies are still in early adoption stages. Only Function-as-a-Service (FaaS) and cloud-native CI/CD tools are seeing substantial use in pilot or production environments.

What other cloud-native technologies are you considering or using?

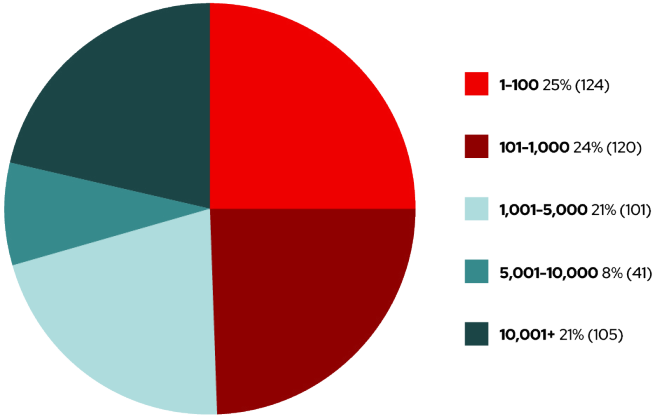


About our respondents—core demographics

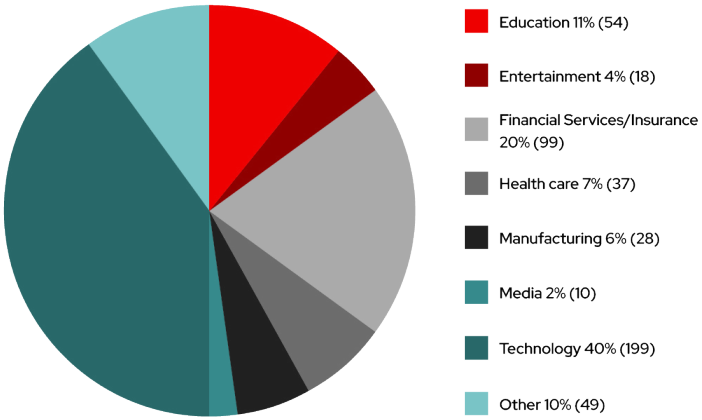
Which area best captures your functional role?



What is your company size?



What industry do you work in?



Learn more about Red Hat Advanced Cluster Security for Kubernetes

Red Hat Advanced Cluster Security for Kubernetes is a Kubernetes-native container security platform that protects your application across build, deploy, and runtime as you progress on your container journey. As your environment grows more complex and you depend on more automation, our platform will let you operationalize security in those more sophisticated environments and keep pace with the speed of DevOps.

Kubernetes-native security provides the following crucial benefits.

- **Minimize operational risk:** Align security with DevOps by using Kubernetes-native controls to mitigate threats and enforce security policies that minimize operational risk to your applications.
- **Reduce operational cost:** Reduce the overall investment in time, effort, and personnel, and streamline security analysis, investigation, and remediation by using a common source of truth.
- **Accelerate DevOps productivity:** Accelerate the pace of innovation by providing developers actionable and context-rich guardrails embedded into existing workflows and tooling that support developer velocity.

Ready to see Red Hat Advanced Cluster Security for Kubernetes in action? Get a personalized demo tailored for your business and needs.

Request demo