StackRox

The State of Container and Kubernetes Security

Winter 2020

with commentary from

451 Research®

StackRox

The State of Container and Kubernetes Security

Winter 2020

Introduction

We're continuing our pioneering research on how companies are meeting the challenges of securing their vital container and Kubernetes environments. This State of Container and Kubernetes Security Report, Winter 2020, is our third edition of the report, this time compiling survey results from more than 540 IT professionals. This edition also uncovered new findings about the impact that security concerns are having on companies embracing Kubernetes and containers.

The most startling data point in the report is the revelation that nearly half of all companies surveyed have delayed moving an application into production because of concerns over security of containers or Kubernetes. Consider the irony here – companies are adopting containers to move fast but they're slowing down application releases because of security concerns.

Another stand-out finding is that nearly everyone – 94% of respondents – admitted to experiencing a security incident in the last 12 months. In many cases, they found a misconfiguration, and only 27% of respondents had a runtime incident, but this data highlights how hard it is to get security right with this tech stack.

We encourage you to benchmark yourself against the findings in this report to determine whether it's time to accelerate your efforts to apply security controls across containers and Kubernetes. Delaying security means delaying revenue and putting the business at risk.

You have lots of security advantages to leverage in containers and Kubernetes – you now need the know-how, tooling, and processes to put those capabilities to work so you can reap all the sizable advantages of running fast in a DevOps, cloud-native world.

One of the most consistent results we get on our own surveys of DevOps and cloud-native security technologies is how important security is for those environments. It is interesting to see how this observation fits well with the StackRox study, highlighting the need for both engineering and security professionals to have visibility and then properly deploy security controls and practices for container and Kubernetes environments."

Fernando Montenegro is a Principal Analyst on the Information Security team of 451 Research

451 Research

Container security concerns are inhibiting business innovation

44% of organizations have delayed deploying apps into production due to security concerns, mitigating the greatest benefit of containerization – agility

Containers, Kubernetes, and microservices application patterns are three of the leading drivers of enterprise IT innovation and digital transformation. Companies have moved quickly to embrace these technologies for their advantages in application development and deployment, from faster development and deployment to quicker bug fixes and patches, leading to faster feature delivery that drives competitive differentiation.

According to our latest survey results, faster application development and release is considered the biggest benefit of containerization, but nearly half of the respondents have had to delay an application roll out because of security concerns.

Companies understand they can't realize the advantages of containers and Kubernetes without getting security right. To see such a large percentage – 44% – acknowledge they've slowed or halted application deployment into production due to security concerns means these companies are not fully benefiting from the primary advantage – faster app delivery – of moving to containers.

Of the following container and Kubernetes benefits,which has benefited your organization the most?

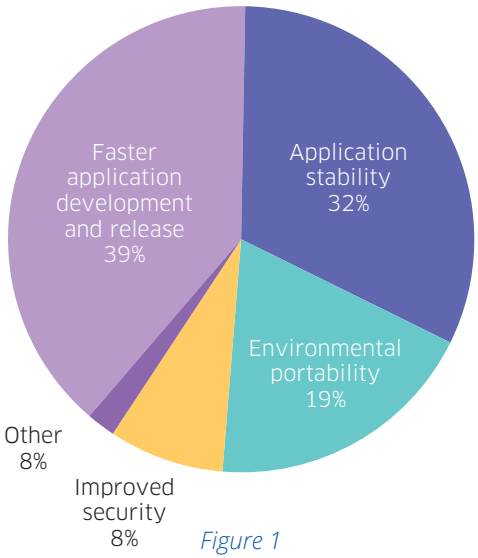

Figure 1

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

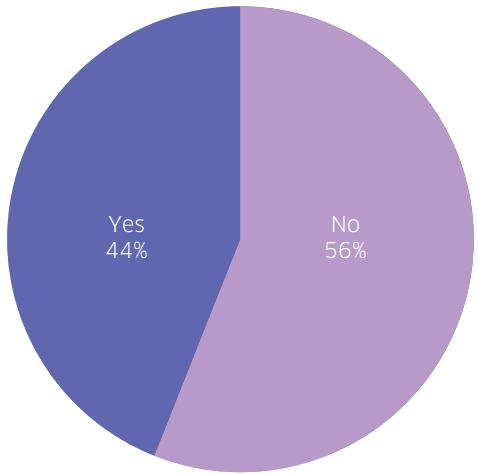

Figure 2

To see such a large percentage – 44% – acknowledge they've slowed or halted application deployment into production due to security concerns means these companies are not are not fully benefiting from the primary advantage – faster app delivery – of moving to containers.

Nearly everyone has experienced a security incident in Kubernetes and container environments

Only 6% did NOT have a security incident in the last 12 months

A whopping 94% of 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 helps explain the previous finding, that nearly half of respondents had delayed an application deployment because of security concerns.

Data breaches and exposures often result from human error, and containers and Kubernetes have a lot of knobs and dials, increasing the chances of human error. Getting all the configurations correct is often very challenging, even for seasoned developers. Not surprisingly, 69% of respondents have experienced a misconfiguration incident in their environments over the last 12 months. Another 27% reported a security incident during runtime, and 24% have had a major vulnerability to remediate.

In the past 12 months, what security incidents or issues related to containers and/or Kubernetes have you experienced? (pick all that apply)

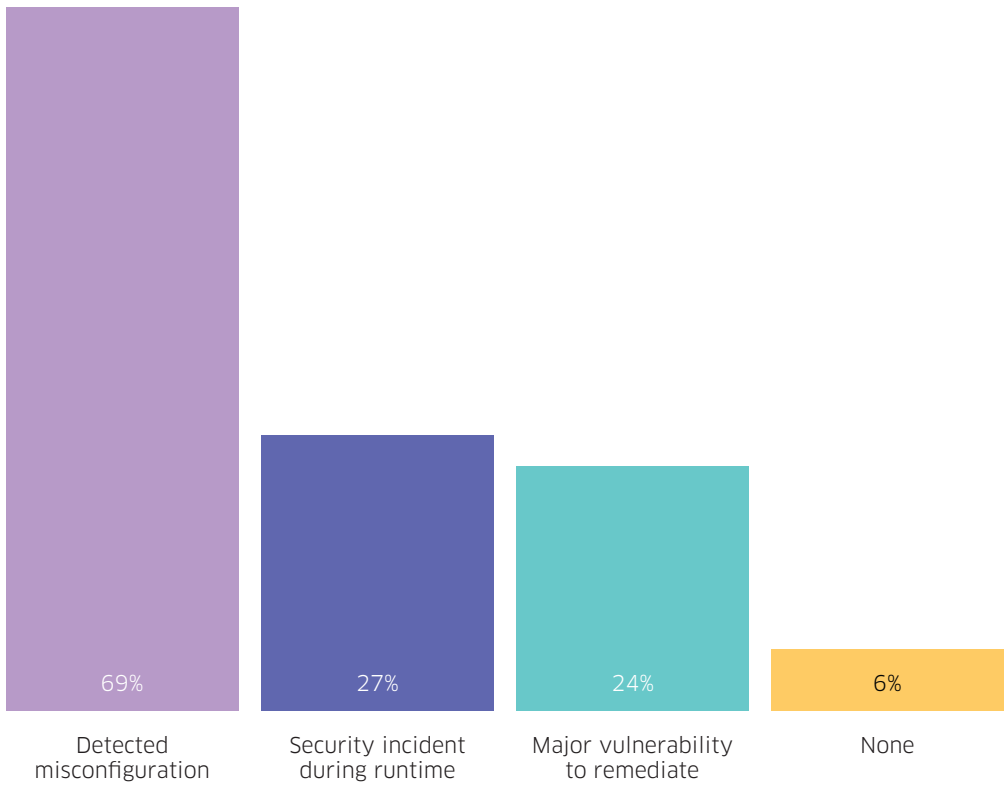

Figure 3

Misconfiguration is the most common security incident

Almost a quarter – 21% – have experienced two or more types of security incidents, with 5% having experienced all three

In our survey, 18% of respondents experienced both a misconfiguration and at least one more security incident during the last 12 months. This combination is critical because a misconfiguration by itself might not be harmful but, when compounded by an exploitable vulnerability, for example, it can pose a much greater risk for a breach.

And while 6% of respondents were lucky enough to not experience any type of security incident, 5% were particularly unlucky, experiencing a misconfiguration, a serious vulnerability, and a runtime threat. The numbers underscore the importance of not only shifting left in applying security controls but also ensuring you enforce security across the full container life cycle.

In the past 12 months, what security incidents or issues related to containers and/or Kubernetes have you experienced? (pick all that apply)

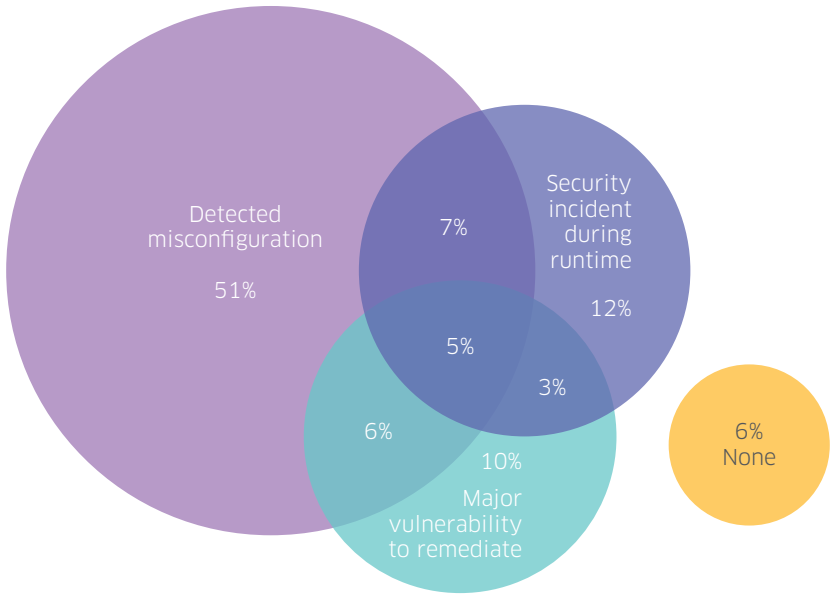

Figure 4

The numbers underscore the importance of not only shifting left in applying security controls but also ensuring you enforce security across the full container life cycle.

Security once again tops the list of concerns with container strategies

Investments in container security continues to fall short of expectations

For the third time in a row in this survey, inadequate investment in security leads the list of concerns users have about their company's container strategy (37%). When combined with not taking the threats to containers seriously (14%), more than half of respondents identify security as their biggest source of concern. Given what we know about the prevalence of security incidents in these environments (94%), it should come as no surprise that security remains the #1 concern.

It is encouraging to see, however, that companies are increasing their container strategy maturity, as seen in the 35% drop in people saying their strategy isn't detailed enough (22% today compared to 34% in Spring 2019). As other findings in this report highlight, organizations are diving headfirst to adopt containers and Kubernetes. If they don't make the necessary investments in security strategies and tooling, they will not only risk the security of their critical applications but also inhibit the business benefits containers bring.

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

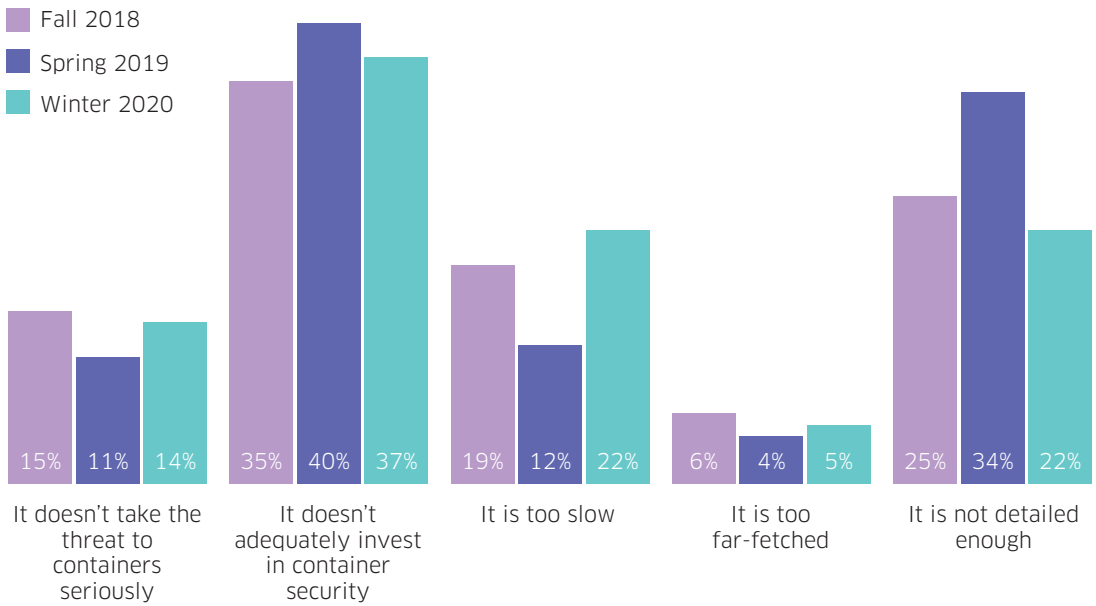

Figure 5

Organizations are diving headfirst to adopt containers and Kubernetes. If they don't make the necessary investments in security strategies and tooling, they will not only risk the security of their critical applications but also inhibit the business benefits containers bring.

Container security strategies continue to mature

Nearly half of respondents have an intermediate or advanced strategy in place

In just the six months since our last survey, organizations have made tremendous progress in advancing their container and Kubernetes security strategies. The number of respondents with an intermediate or advanced strategy jumped from 41% to 48%, marking a growth rate of 17%. Even more promising is the drop in respondents lacking any form of security strategy; that number has dropped from 19% to 6% in the past six months, a 68% decrease.

This data is promising, but the previous finding shows that despite maturing security strategies, organizations must invest more in Kubernetes and container security.

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

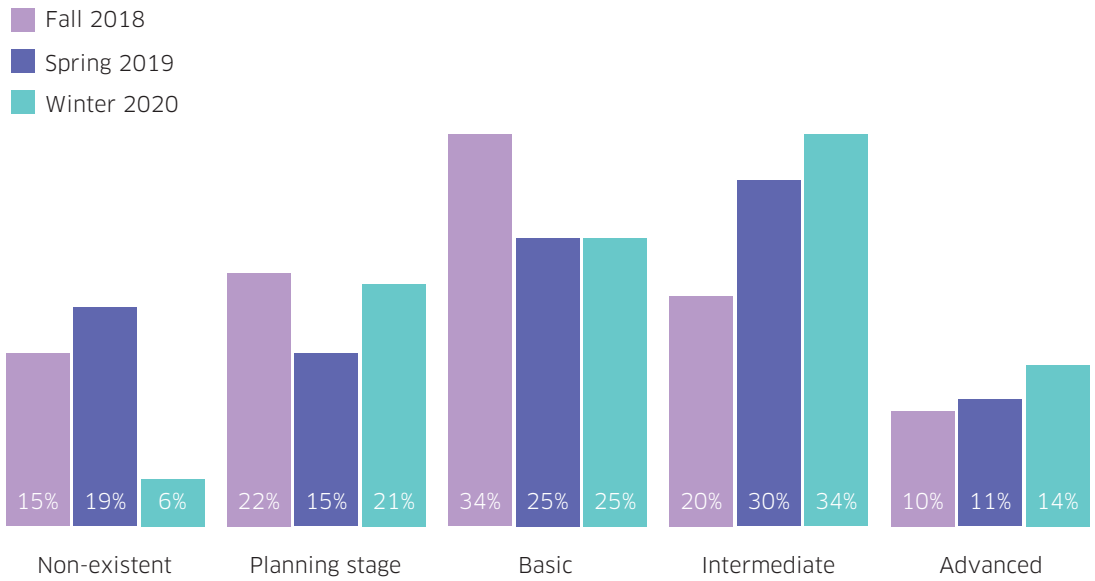

Figure 6

Misconfigurations continues to pose the greatest security risk – to no one's surprise

The prevalence of misconfigurations underscores the need for automating configuration management

For the third time in a row, survey respondents identified exposures due to misconfigurations as the most worrisome security risk in their container and Kubernetes environments (61%) – more the twice the level of concern over vulnerabilities (27%) and more than five times that of attacks (12%).

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 know not to expose the Kubernetes dashboard to the Internet, but network segmentation via YAML files and Kubernetes RBAC are just two examples of more challenging settings to get right.

One way to address this challenge is to automate configuration management as much as possible, so that security tooling – rather than a human – provides the guardrails for developers and DevOps teams to configure containers and Kubernetes securely.

Of the following risks, which one are you most worried about?

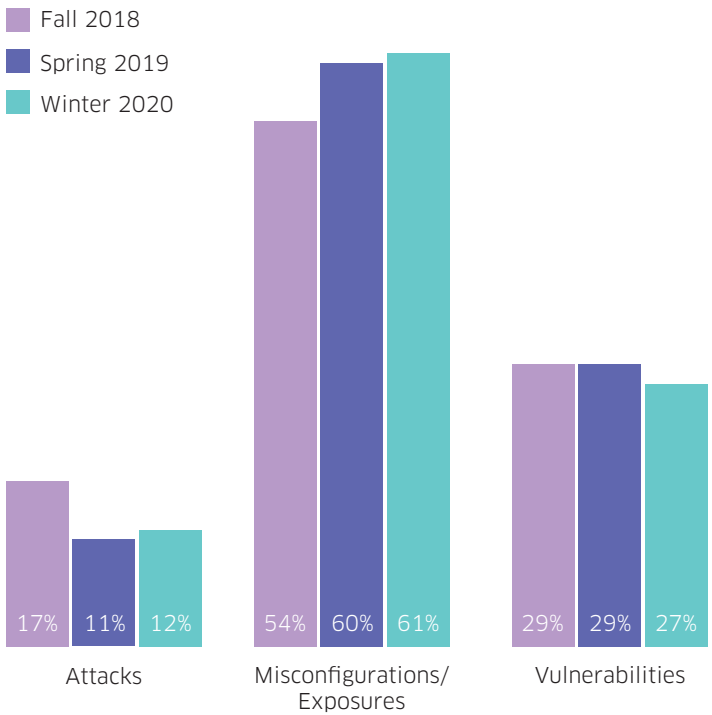

Figure 7

Automate configuration management as much as possible, so that security tooling – rather than a human – provides the guardrails for developers and DevOps teams to configure containers and Kubernetes securely.

Runtime continues to be the life cycle phase respondents worry about the most

Runtime is once again the container life cycle phase that organizations worry about the most. This finding is perplexing at first glance, given that an overwhelming majority of respondents identify misconfigurations as the source of biggest security risk (*Figure 7*) and have experienced a misconfiguration incident more often than other type (*Figure 3*). However, the data makes more sense when you consider that issues during runtime feel like they're less under your control and that other security mis-steps, such as a misconfiguration, will lead to a security problem only during runtime, not build or deploy.

Worries about the build stage have fallen over the past two surveys, dropping by 32% in the last six months. One possible explanation is that as security strategies mature and organizations adopt a shift left approach, starting with image scanning and vulnerability management, worries about the build phase naturally decrease.

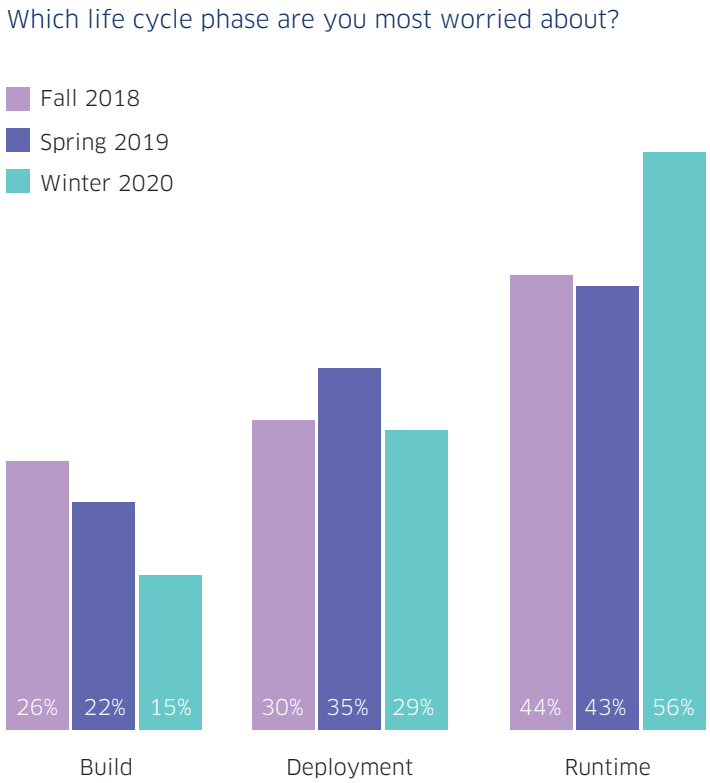

Figure 8

Organizations are diving head-first into containerizing their applications

Data from this survey shows a great surge in containerization of apps. Since our last report six months ago, the percentage of organizations that have containerized more than half of their applications has risen from 23% to 29%, a growth rate of 22%. At the same time, the number of organizations that have containerized fewer than 10% of their apps fell from 32% to 21%.

What percentage of your apps are containerized today?

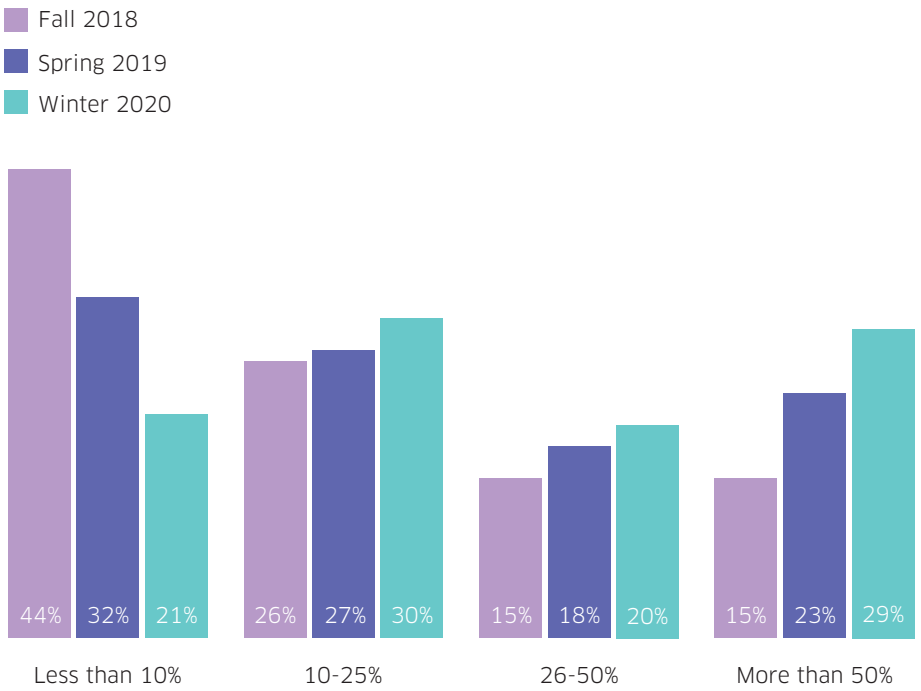

Figure 9

Organizations have far more containerized applications in production today than Spring 2019

Not only are organizations containerizing more of the apps – they're also running more of those containerized apps in production. The percentage of organizations with more than half their containers running in production jumped from 22% to 29% in the past six months, a growth rate of 32%. In the same time period, organizations running fewer than 10% of their containers in production has fallen from 39% to 28%.

The percentage of organizations with more than half their containers running in production jumped from 22% to 29%.

What percentage of your containerized apps are running in production?

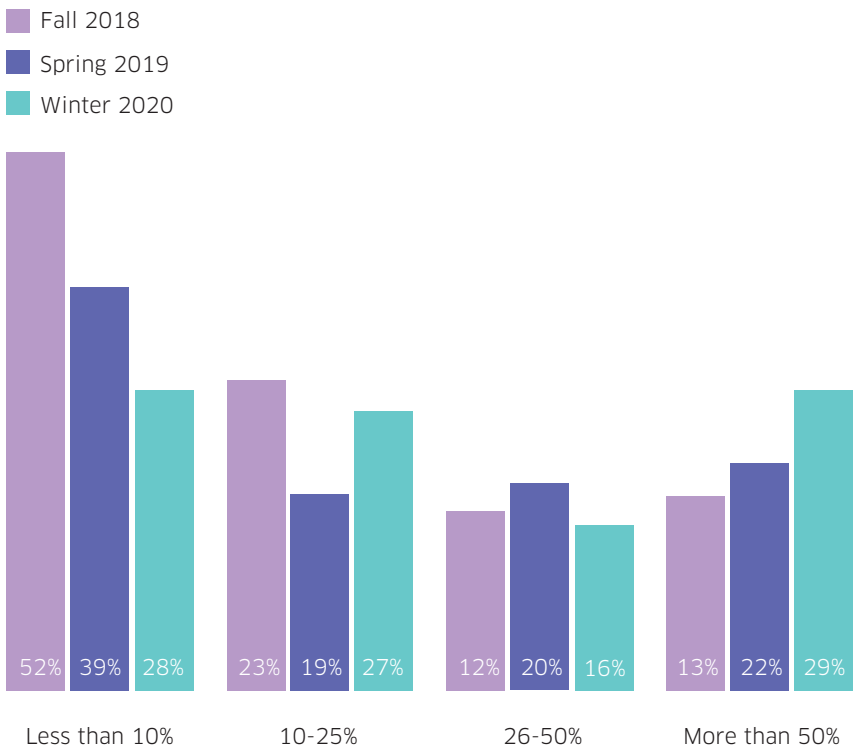

Figure 10

Hybrid deployments dropped while cloud-only grew

Multi-cloud deployments grew again but trail single-cloud deployments by a sizable margin

Industry chatter about adopting a multi-cloud approach runs high, but actual deployments across multiple cloud providers still lags single-cloud deployments – 51% of respondents run their containers in a single cloud vs. 35% who run in multiple public clouds. When isolating cloud-only container deployments, the delta is even greater, at 27% in multi-cloud and 13% in a single cloud.

Overall, the hybrid model continues to be the most popular architectural approach to deploying containers, with 46% running both on-prem and in the cloud. Those running with cloud-only deployments stand at 40%, and on-premises only deployments have fallen to a new low of 14%, down from 31% in Fall 2018.

With hybrid models continuing to be the dominant approach, organizations need security that runs the same way – a Kubernetes-native container security platform delivers environment-agnostic control.

With hybrid models continuing to be the dominant approach, organizations need security that runs the same way – a Kubernetes-native container security platform delivers environment-agnostic control.

Where do you have containers running?

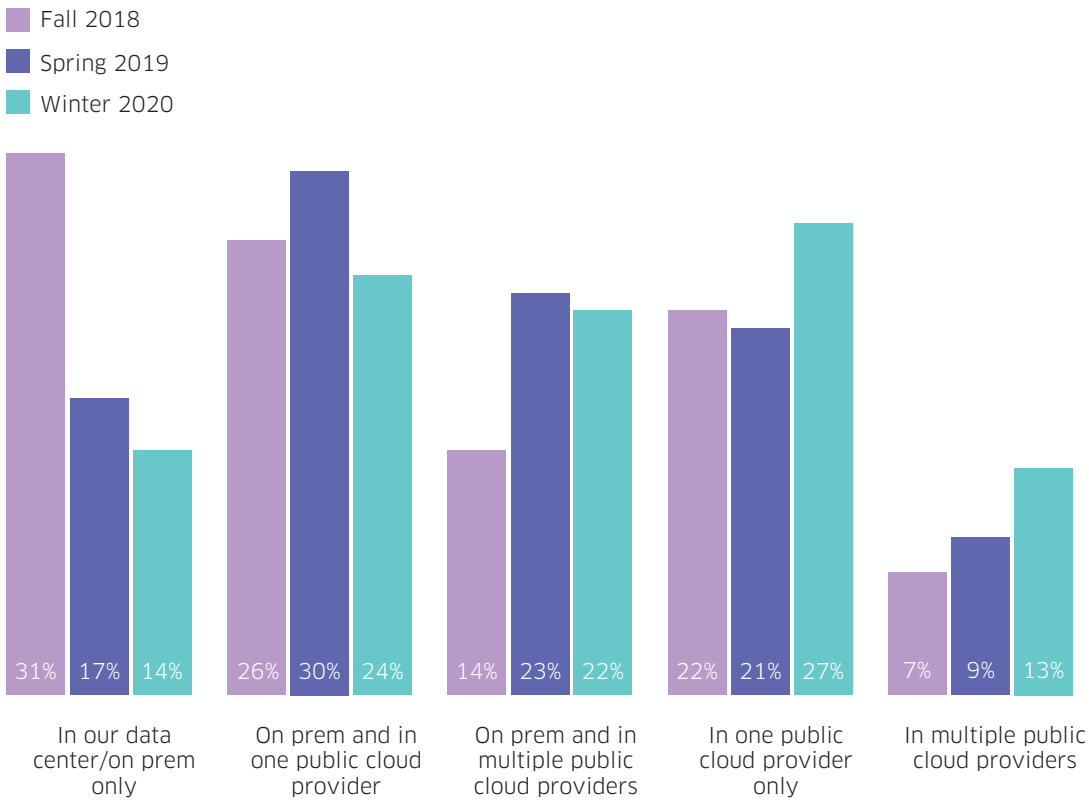

Figure 11

AWS vs. Azure vs. GCP

AWS continues to dominate, but the race for #2 heats up

As expected, Amazon Web Services (AWS) continues to dominate the container market, but the race for second has tightened. While Microsoft Azure remains in second place, Google Cloud Platform (GCP) has grown its third-place standing from 28% in Spring 2019 to 35% today. That GCP rivals Azure so closely might not be surprising, since Google created Kubernetes before donating it to the Cloud Native Computing Foundation (CNCF). Also, Google Kubernetes Engine (GKE) is one of the most feature-rich managed Kubernetes service in the market, especially in the area of cluster management – again, in large part due to Google's deep experience orchestrating containers at scale.

If you're running containers in the public cloud, which provider(s) are you using? (pick as many as apply)

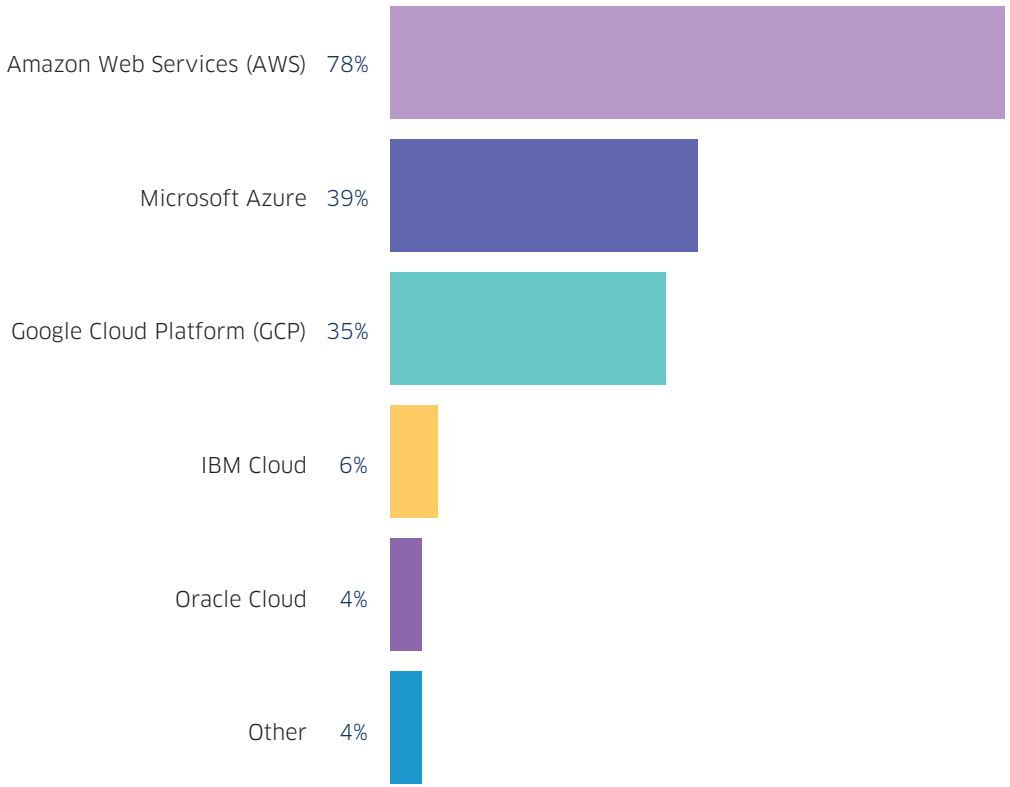

Figure 12

A deeper look at Azure vs. GCP users

Container maturity for Azure users lags both GCP and the overall population

Azure users tend to be earlier in their container adoption journey than organizations running on AWS or GCP. Only 20% of Azure users have containerized more than half of their applications, significantly lower than the 33% across all non-Azure users. Correspondingly, fewer Azure users have more than half their containerized apps deployed in production environments – 22% vs. 34% for all non-Azure respondents.

In contrast to Azure users, GCP users are ahead of the curve in their container adoption. Nearly a third (31%) of GCP users have containerized more than half their applications, which is slightly higher than the 27% across all non-GCP respondents. Likewise, a great number of GCP users have more than half their containerized apps deployed in production environments – 31% vs. 29% for non-GCP users.

What percentage of your apps are containerized?

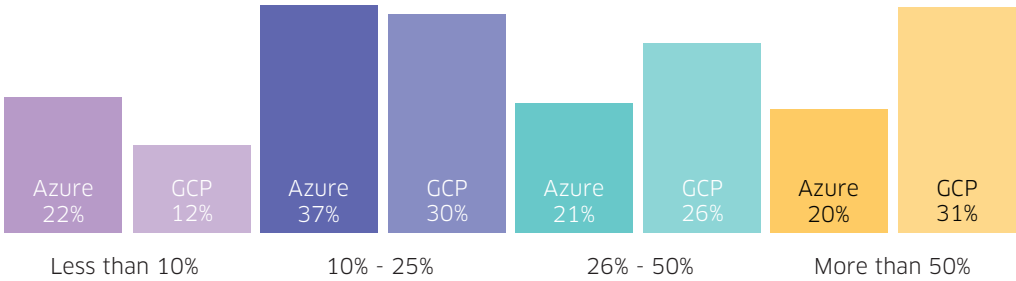

Figure 13

What percentage of your containerized apps are running in production today?

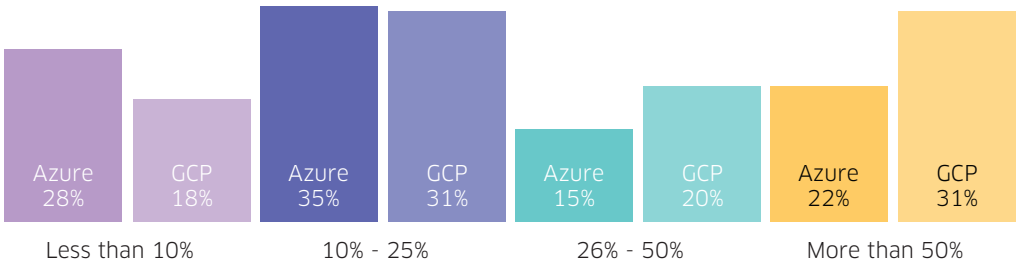

Figure 14

Kubernetes continues to dominate the container orchestration market

Managed Kubernetes offerings from cloud providers see massive growth in adoption

Our survey reinforced the dominance Kubernetes enjoys in the market, with 86% of respondents using it for container orchestration. But how they're deploying Kubernetes has changed dramatically. Self-managed Kubernetes had been the most dominant approach, but it now lags Amazon EKS (barely, at 35% vs. 37%). The self-managed approach dropped 20% while EKS climbed 37%, AKS increased 31%, and GKE grew a whopping 75% compared to Spring 2019.

Along with the self-managed approach, AWS Elastic Container Service (ECS) and Docker Enterprise Edition - Kubernetes also both shrank in market share. ECS enjoys a hold-over effect as one of the pre-Kubernetes orchestration services, though we expect it to continue to lose ground to EKS. As for Docker, losses there could easily be tied to uncertainty about the technology's future given the sale of Docker's enterprise product line to Mirantis.

Percentage of survey respondents using Kubernetes:

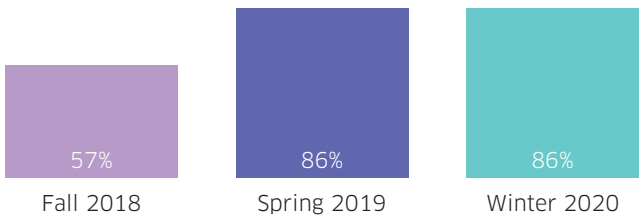

Figure 15

Which of the following container orchestrators do you use? (pick as many as apply)

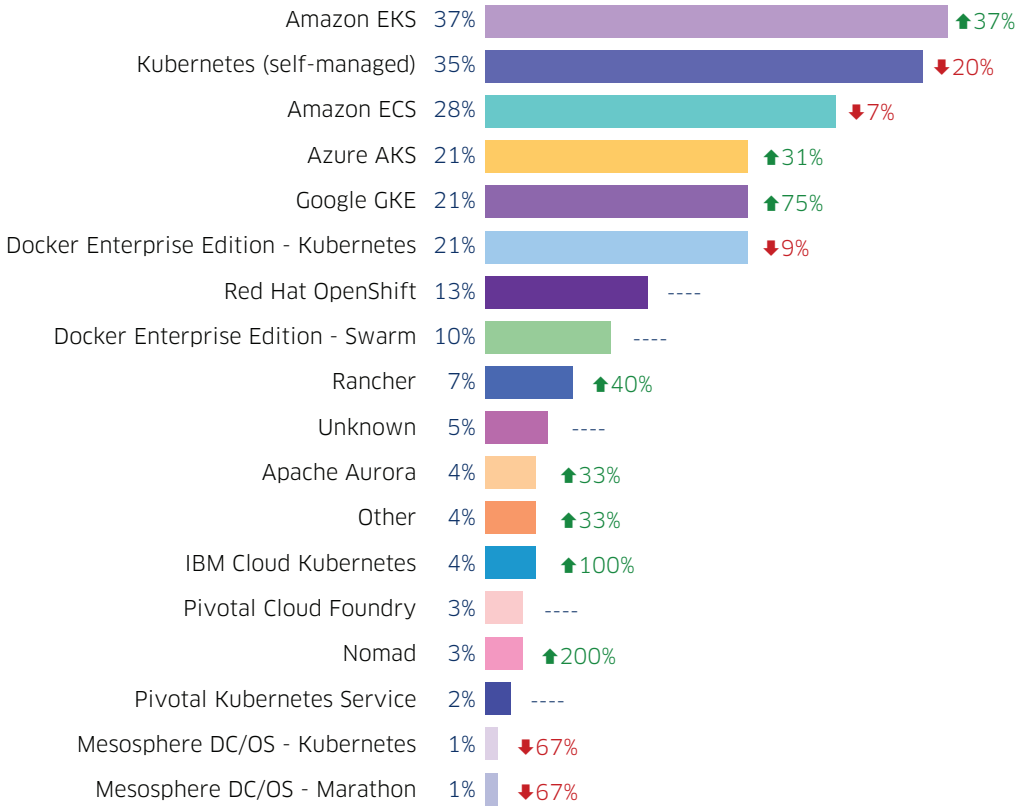

Figure 16

Continued skills shortage among Kubernetes users likely to benefit managed Kubernetes service providers

As with any emerging technology, Kubernetes is suffering from a skills gap. Organizations cited both an internal skills shortage and a steep learning curve as the two most significant Kubernetes challenges impacting their company. More than half of the respondents cited these problems, and we expect these difficulties are likely tied to the growth in adoption of managed Kubernetes services.

Of the following Kubernetes challenges, which has impacted your organization the most?

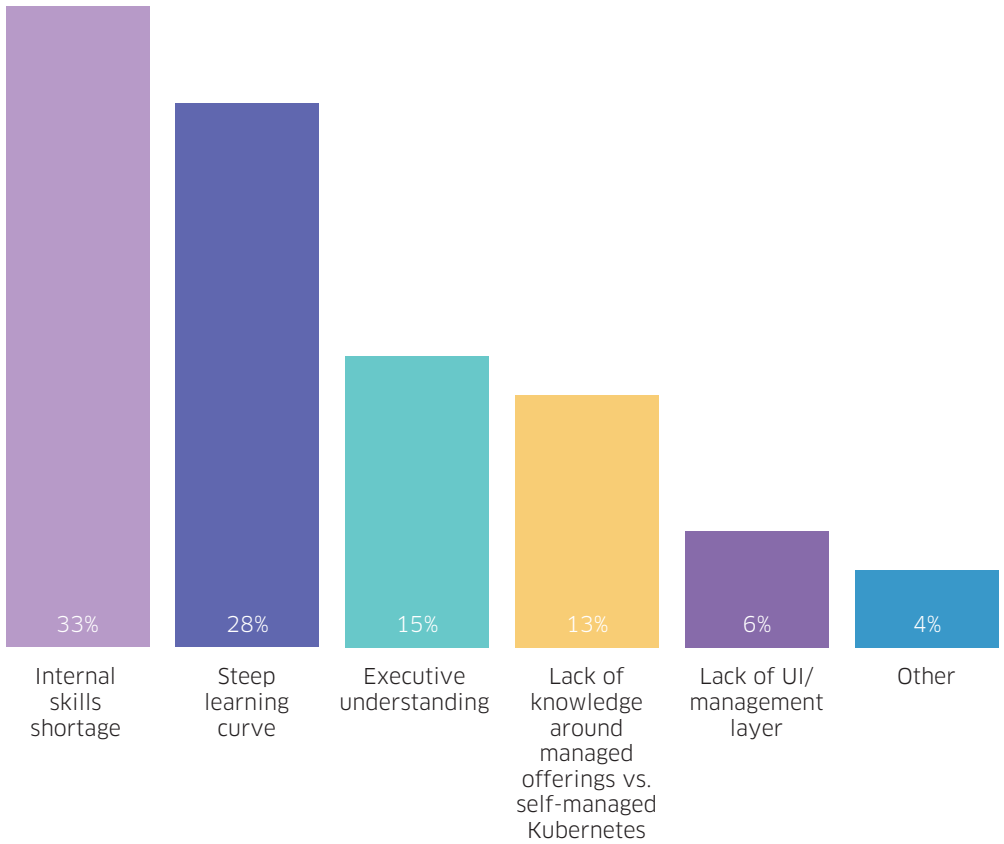

Figure 17

Respondents expect a very feature-rich container and Kubernetes security platform, with emphasis on “shift-left” features

Respondents expect a lot out of their Kubernetes security platforms – more than half of the respondents cited each of seven main use cases 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 across developers, DevOps teams, and Security teams.

Respondents put a premium on capabilities that allow them to shift security left, with vulnerability management and configuration management taking two of the top three spots. Compliance is also seen as a critical feature, despite the relatively low percentage of respondents (only 20%) identifying as being in a security role.

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

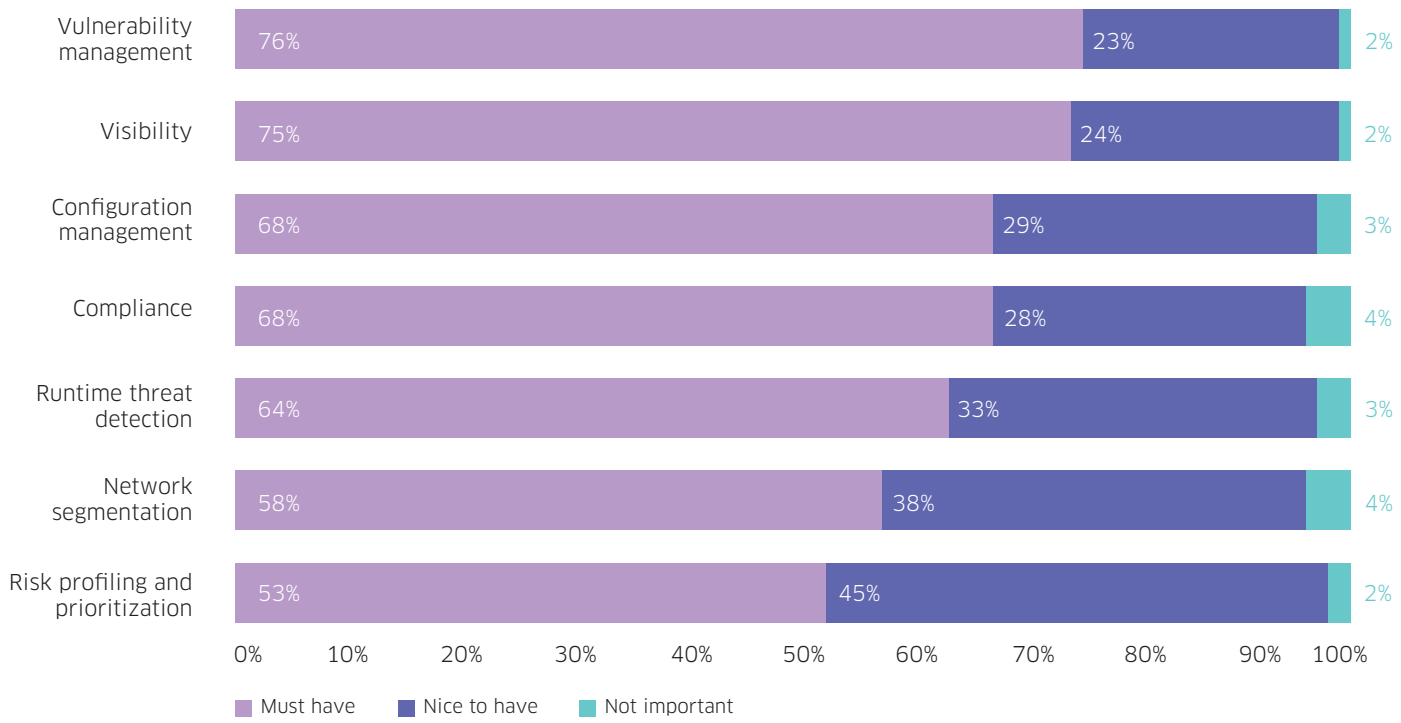

Figure 18

Top use case requirements map to DevOps taking lead in managing container security

Organizations are increasingly looking to DevOps to take lead in securing containerized applications

Across all operations roles, DevOps is considered the party most responsible for managing container security, with 81% of respondents selecting that role, followed by security at a distant 51%. These findings map well to what respondents consider must-have security capabilities (*Figure 18*).

However, when it comes to container and Kubernetes security, it takes a village. We see considerable overlap between different roles considered responsible for security. Of the 81% that believe DevOps should take lead in managing container security for example, nearly half of those same respondents (48%) also selected security (note that the question allowed for multiple selections). Figure 20 shows the extent to which security responsibility is shared among different teams and roles. Container and Kubernetes security tooling must therefore facilitate close collaboration among different teams – from Security to DevOps to Ops and Developers – instead of perpetuating the silos that often plague organizations.

When it comes to container and Kubernetes security, it takes a village. Container and Kubernetes security tooling must therefore facilitate close collaboration among different teams – from Security to DevOps to Ops and Developers – instead of perpetuating the silos that often plague organizations.

Who will be responsible for operating a container security platform? (pick as many as apply)

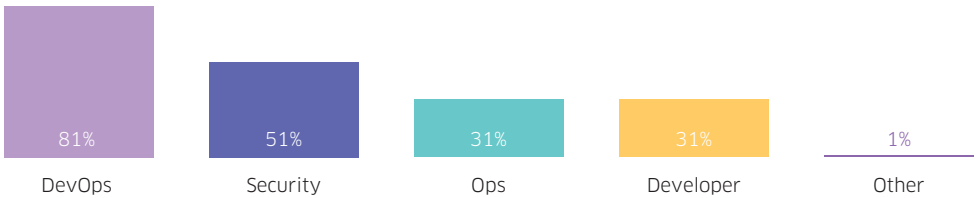

Figure 19

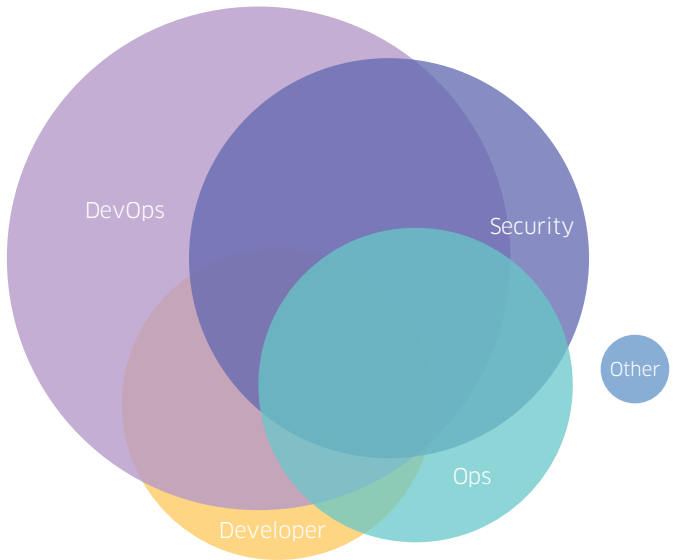

Figure 20

The cloud-native infrastructure requires and facilitates closer collaboration

The good news in solving the Kubernetes and container security challenge is that DevOps and security teams are already working together more closely. This trend makes sense, given that Kubernetes itself helps unify what had been very separate disciplines of infrastructure and security. With Kubernetes, controls are part of the infrastructure, enabling organizations to make progress on instantiating "security as code." However, teams need workflows, processes, and next-generation security tooling that foster this cooperation across groups.

How are containers changing how DevOps and Security work together? (pick as many as apply)

Figure 21

Implications for containers and Kubernetes security

The findings in this survey of 541 respondents make clear that organizations are putting at risk the core benefit of faster application development and release by not ensuring their cloud-native assets are built, deployed, and running securely. With the prevalence of misconfigurations 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 nearly half of our respondents delaying going into production because of security concerns, clearly a lack of security is inhibiting business acceleration and innovation.

1. Leverage Kubernetes-native architectures and controls.

If anyone still thought Kubernetes was perhaps a fad, that idea can now firmly be put to rest. Organizations are rapidly standardizing on Kubernetes, which provides rich data and native controls that enable the most robust and portable security. The context Kubernetes provides about how containers and deployments are configured and running will enrich your understanding of risk in your environment. Leveraging Kubernetes for admission control, network segmentation, scaling non-compliant services to zero, and killing infiltrated pods will enable far better enforcement than layering in separate proxies or shims. Further, that approach will ensure that DevOps and Security share a common source of truth for closer collaboration.

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, we have the opportunity to position security as a business accelerator, by helping developers build good 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 security posture, so developers can focus on feature delivery.

3. Require portability across all Kubernetes environments.

With most organizations deploying containers in both on-prem and public cloud environments (frequently in multiple clouds), security must apply consistently wherever your assets are running. The common element is Kubernetes, so make Kubernetes your source of truth, your point of enforcement, and your universal visibility layer so you can enable consistent security. Managed Kubernetes services may jump-start 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 DevOps or DevSecOps teams to run container security platforms, your security tooling must help bridge security and DevOps. To be effective, the platform must enable security to apply 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 will net you nothing. Identifying for that developer the three riskiest deployments, and showing why they're risky, will get you action that will genuinely improve your security posture.

About our respondents – technologies being adopted

The prominence of the Docker runtime engine has never been more prevalent. Only containerd managed to show more than 10% penetration.

What container runtime(s) do you use? (pick as many as apply)

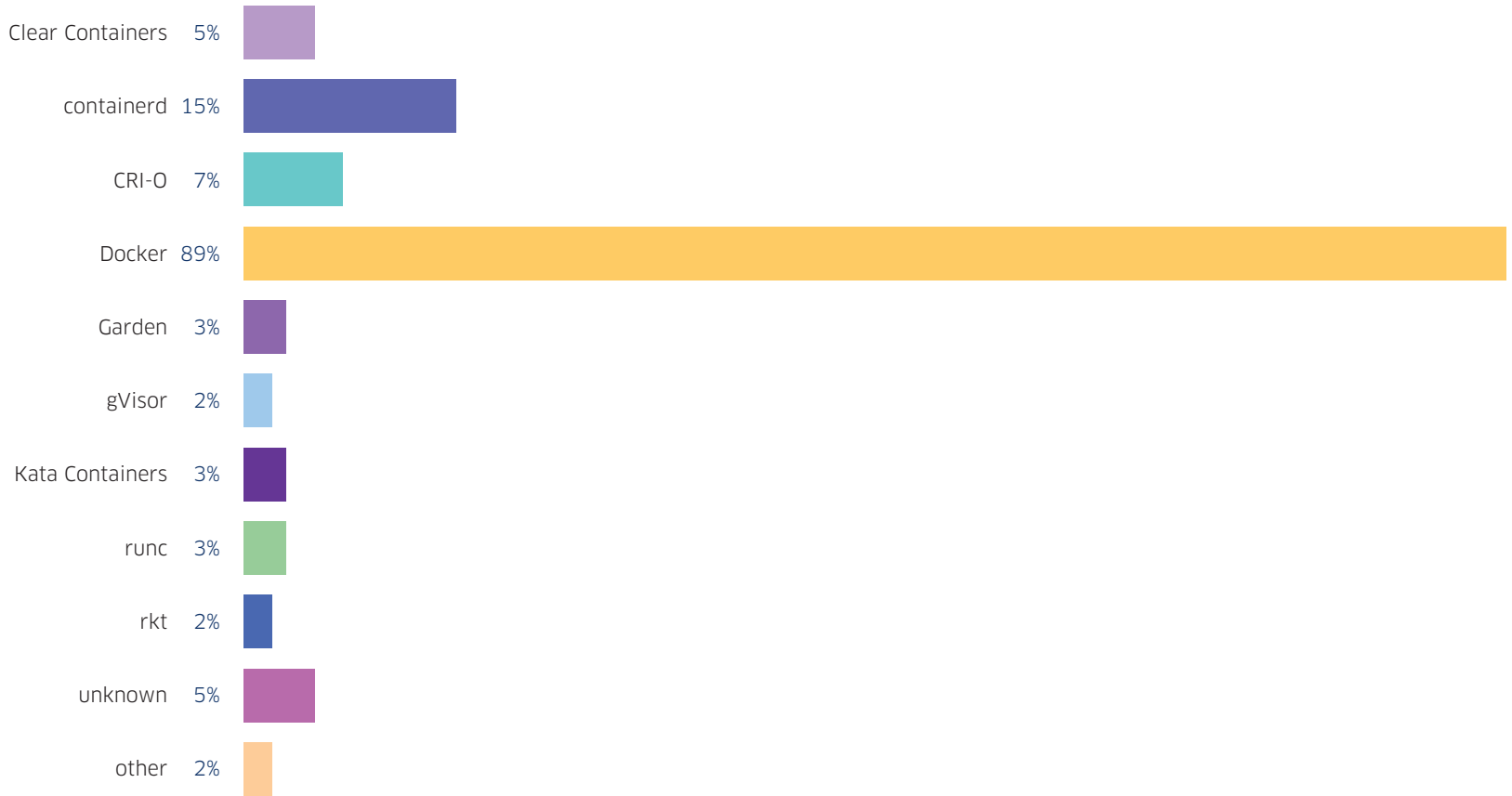

Figure 22

About our respondents – technologies being adopted

Emerging cloud-native technologies are still in early adoption stages, with most respondents either investigating or not interested at all. Only Function as a Service on AWS and Kubernetes-native Function as a Service are seeing significant use in pilot or production environments.

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

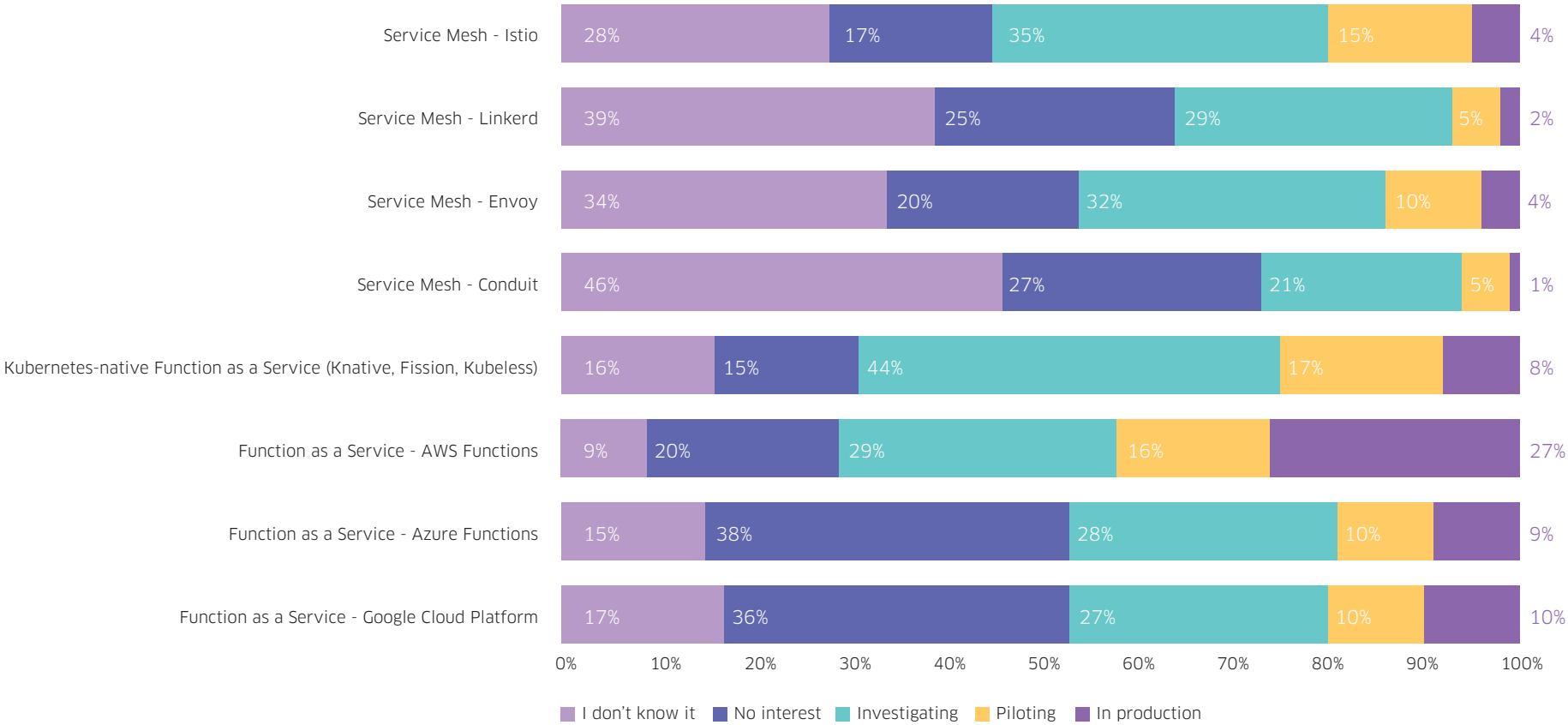

Figure 23

About our respondents – technologies being adopted

Organizations continue to containerize old and new apps at fairly even rates, which highlights the fact that containers are just as applicable to legacy applications as new ones.

When you think of containerized applications, you might think of next-gen, microservices-based apps. However, much like the results from six months ago, we continue to see organizations containerize legacy applications. In a move reminiscent of the old "lift and shift" days in the cloud, these organizations are simply taking existing code and putting it into containers – almost half the respondents are containerizing older apps without making any code changes.

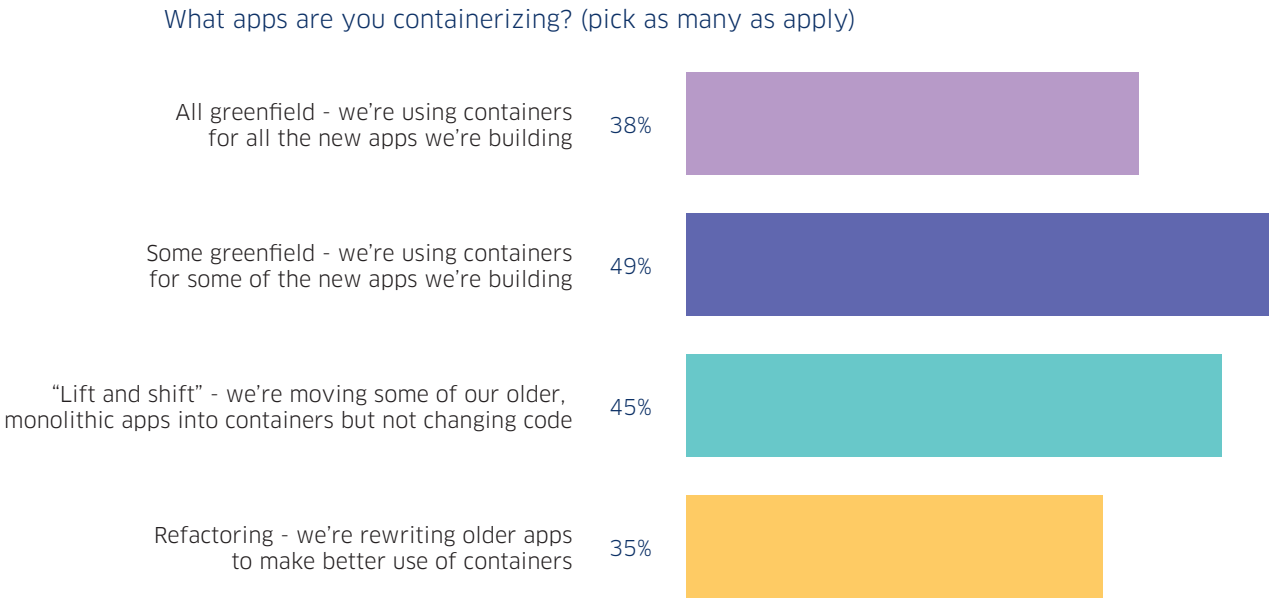

Figure 24

About our respondents – core demographics

More than 540 IT decision makers shared their perspectives for this third edition of the industry's only report on the State of Kubernetes and Container Security. A quarter of the respondents identify security or compliance as their primary IT role, and almost 40% work in companies of more than 5,000 employees.

Running containers is often associated with cloud-native companies, but many Global 2000 organizations have adopted containers to maintain their competitive edge – companies with more than 10,000 employees constitute nearly a third of our respondents.

It's no surprise that high-tech and financial services companies dominate our survey responders — high-tech companies typically adopt the tech they create, and financial services companies either are next-gen fin-tech companies themselves or need to innovate fast to keep pace with them.

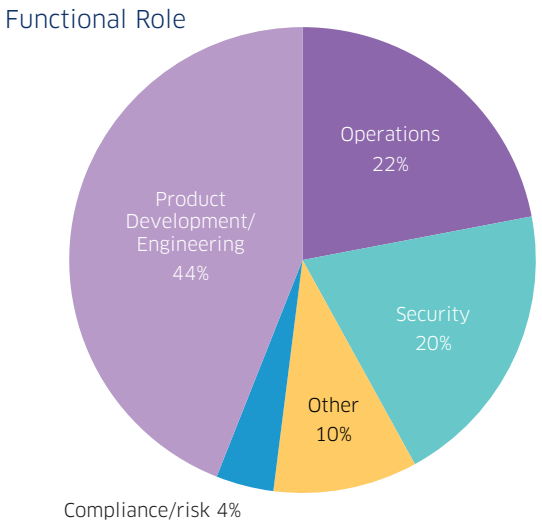

Figure 25

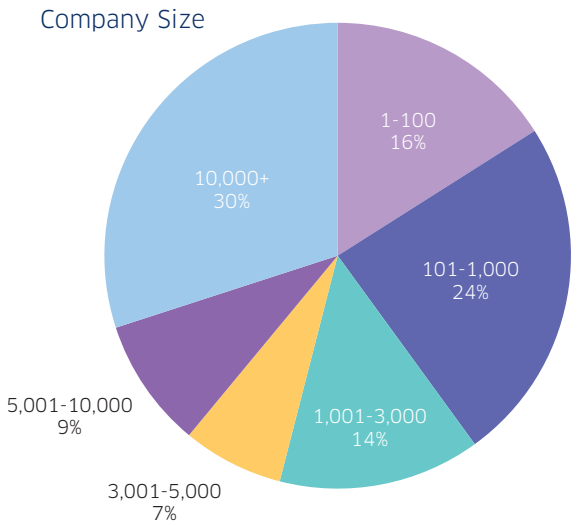

Figure 26

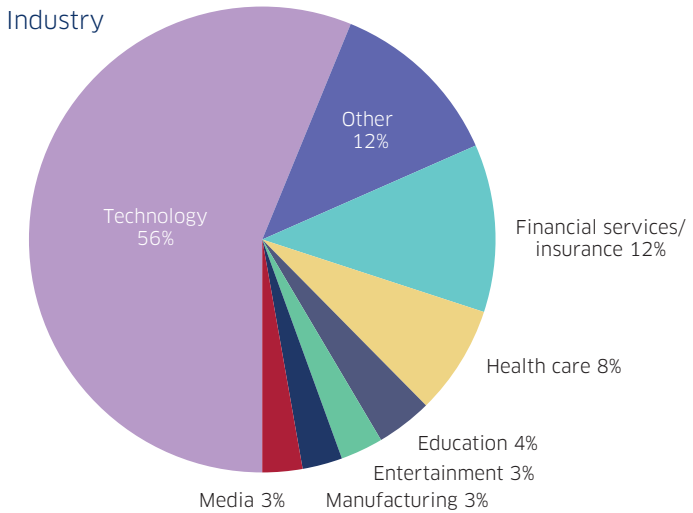

Figure 27

Top seven container security use cases for Kubernetes environments

Companies are rapidly moving their container workloads to production to accelerate feature velocity and drive digital transformation and business growth. Our data shows that organizations have standardized on – or are on the path to standardize on – Kubernetes, and its rapid adoption offers equal parts promise and peril. Promise, in the form of infrastructure that enable far greater inherent security than ever before. And peril, as companies struggle to overcome a skills gap and configure the technology in the most secure manner. Container workloads – often deployed across varying environments (cloud, on-prem, hybrid) – require a security approach that is portable; protects the full life cycle; and leverages Kubernetes' rich context, native controls, and scalable policy enforcement to build a bridge between DevOps and Security.

The following seven container and Kubernetes security use cases top the list of priorities for most organizations – follow the best practices gleaned from your peers and shared here to get your organization on the right track.

1. Vulnerability management

Most organizations start with vulnerability management – the challenge is to quickly move beyond the limited value provided by image scanning. Organizations must also identify vulnerabilities in Kubernetes, and they need a way to quickly pinpoint newly discovered vulnerabilities in already running deployments. Start with vulnerability management, but demand more than image scanning for this use case.

2. Visibility

As the second most often-cited security use case, visibility into your container and Kubernetes environments is at the root of being able to properly secure that environment. Only when your security tooling is fully embedded into Kubernetes can you understand your cloud-native infrastructure, including images, containers, pods, namespaces, clusters, and network policies. You need insights into how each is configured and whether they're compliant with industry standards and your internal security policies.

Top seven container security use cases for Kubernetes environments

3. Configuration management

Misconfiguration poses the greatest security risk to containers and Kubernetes. In today's DevOps-driven environment, configuration management must be as automated and streamlined as possible for it to not slow down application development and deployment. It should be comprehensive, covering containers, Kubernetes, and all their configurable components, including:

- RBAC
- Secrets
- Network policies
- Privilege levels
- Resource limits/requests
- Read-only root file systems
- Annotations, labels
- Sensitive host mount and access
- Image configuration, including provenance

4. Compliance

DevOps moves fast and relies on automation for continuous improvement, so organizations need a compliance solution built to complement – not inhibit – the pace of business. You need to not only adhere to industry compliance requirements but also show evidence of that compliance. You should be able to show which clusters, nodes, or namespaces are compliant with all the individual controls relevant in container and Kubernetes environments from frameworks including CIS benchmarks for Docker and Kubernetes, PCI, HIPAA, and NIST SP 800-190. And it should be dead simple to run on-demand compliance checks and export evidence of compliance.

5. Runtime threat detection

According to our report, runtime is the life cycle phase that customers are most worried about. The security goal in this phase is to detect and respond to malicious activity in an automated and scalable way while minimizing false positives and alert fatigue. Kubernetes offers rich declarative data around images and deployments that delivers valuable context when assessing runtime behavior. Leverage this context to more accurately differentiate between simple anomalies and true threats, and use Kubernetes-native enforcement capabilities to mitigate runtime threats in the most automated and scalable manner.

Top seven container security use cases for Kubernetes environments

6. Network segmentation

Containers pose a unique networking challenge because containers communicate with each other across nodes and clusters (east-west traffic) and outside endpoints (north-south traffic). Kubernetes provides built-in capabilities that enable network segmentation. Leverage those native controls to ensure consistent, portable, and scalable network segmentation regardless of your CNI plugin or Kubernetes distribution. Using the segmentation inherent in Kubernetes ensures that security and DevOps see and act on a single source of truth and consistent information to restrict access and reduce the blast radius.

7. Risk profiling and prioritization

In sprawling Kubernetes environments, manually triaging security incidents and policy violations is time consuming and prone to exacerbating alert fatigue. A better approach is to use the declarative contextual data in your Kubernetes environment to assess risk across all your deployments. Instead of looking at vulnerabilities and CVSS scores alone, understand true risk in your environment and prioritize which security issues should be fixed first. As an example, a deployment containing a vulnerability with severity score of 7 or greater should be moved up in remediation priority if that deployment contains privileged containers and is open to the Internet but down if it's in a test environment and supporting a non-critical app.

Request a demo today!
info@stackrox.com
+1 (650) 385-8329
www.stackrox.com