

Sysdig 2023 Cloud-Native Security and Usage Report



Table of Contents

- 01 Executive Summary 03
- 02 Organizations Struggle to Manage Supply Chain Risk 05
- 03 Zero Trust: Lots of Talk, Little Action..... 10
- 04 Mature Organizations Are Proactively Testing Their Security Posture..... 13
- 05 Millions Wasted on Unused Kubernetes Resources..... 16
- 06 Usage Trends and Insights 17
- 07 Methodology..... 27
- 08 Conclusion..... 28

01

Executive Summary

For the past six years, we have shared an analysis of our real-world customer data to provide the community with insight into changing container usage and security trends. This report is based on data gathered from billions of containers, thousands of cloud accounts, and hundreds of thousands of applications that our customers operated over the course of the last year. This allows us to report on many different aspects of actual usage of containers and cloud, rather than rely on survey results.

The two biggest cloud security risks continue to be misconfigurations and vulnerabilities, which they are being introduced in greater numbers through software supply chains. We dove deep into this data for the 2023 issue of the report because it lands on the priority list of all security leaders. Unfortunately, 87% of container images running in production have a critical or high severity vulnerability. Despite increased adoption of shift-left security strategies to assess code early and often, organizations need runtime security. This is evidenced by the tremendous growth in the adoption of technologies like Falco, a Cloud Native Computing Foundation (CNCF) open source project, that helps organizations detect runtime threats across clouds, containers, hosts, and Kubernetes environments.

Our findings provide signs of hope for overburdened developers, as the data showed opportunities to focus remediation efforts on vulnerable packages loaded at runtime. Only 15% of high or critical severity vulnerabilities with an available fix are actually in use at runtime. Prioritization based on filtering by in use packages enables teams to significantly reduce cycles spent chasing an endless pile of vulnerabilities.

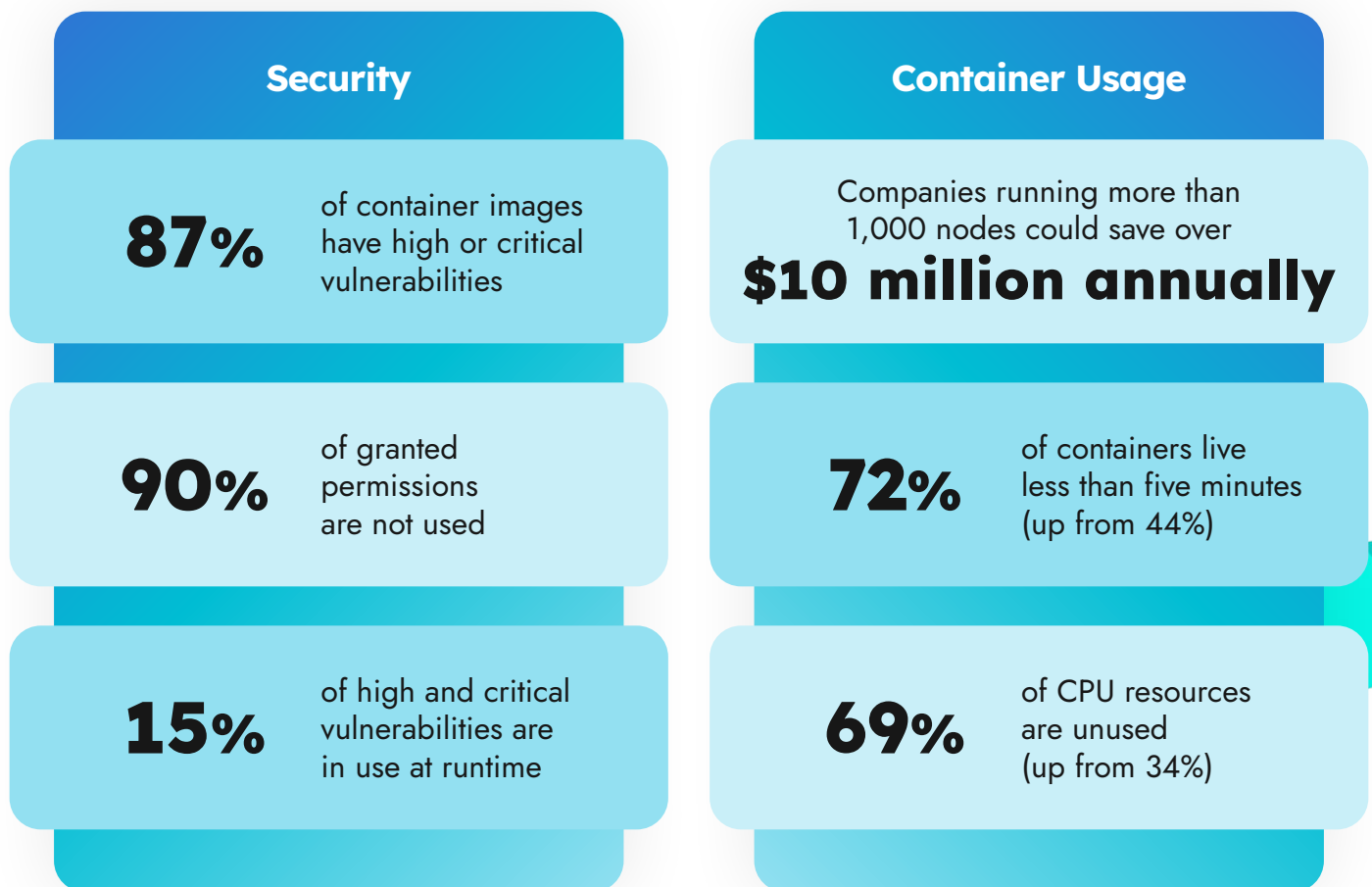
Security leaders voice zero trust as a top priority; however, our data showed that least privilege access rights, an underpinning of zero trust architecture, are not properly enforced. In fact, we found 90% of granted permissions are not used, which leaves many opportunities for attackers who steal credentials. Teams need to enforce least privilege access, and that requires an understanding of which permissions are actually in use.

Given the current macroeconomic challenges, many IT teams are researching ways to reduce cloud costs. Getting accurate utilization and cost information about Kubernetes deployments, or rightsizing, is a tough challenge due to the ephemeral nature of these environments. This report shows over 72% of containers live fewer than five minutes, which is up drastically from previous years of our analysis.

When you combine short lifespans with cluster density growing again this year, it's apparent that organizations will need to look for ways to rein in costs. Our data shows that organizations of all sizes likely have cost overages in their Kubernetes environments, with the largest deployments leaving more than \$10 million on the table.

It brings us great pleasure to present the Sysdig 2023 Cloud-Native Security and Usage Report. This information can be useful for determining the real-world state of security and usage for container and cloud environments. The data can also help inform cybersecurity strategies and priorities. We are confident these insights can help teams, regardless of their company's size or stage in their cloud journey.

Key Trends



02

Organizations Struggle to Manage Supply Chain Risk

Attacks targeting the [software supply chain](#) are on the rise. SolarWinds raised awareness about the risk. More recent events, like the Federal Civilian Executive Branch (FCEB) agency breach, amplify the concern. In the FCEB breach, the Iranian government exploited the Log4Shell vulnerability to deploy a cryptominer, steal credentials, and maintain persistence in the FCEB environment. Development teams increasingly rely on open source software and third-party code, and with that comes the risk of exposure to both known and unknown security vulnerabilities.

“The number and complexity of vulnerabilities can be overwhelming in cloud-native environments. Our developers and security teams have a tiered approach to reviewing and triaging the actual risk of vulnerabilities. By looking at the vulnerabilities that pose a greater risk, meaning they are considered high or critical severity and/or have an active public exploit, we’re able to prioritize effectively and focus on taking actions that matter.”

- **Michael Bourgault,**
IT Security Manager, Arkose Labs

87% of images have high or critical vulnerabilities

High-profile vulnerabilities and exploits, such as Log4Shell and Text4Shell, along with increased guidance from government organizations regarding cybersecurity, have caused many teams to heighten their focus on application security testing. Even

with these high-profile vulnerabilities, there is little evidence of real progress in addressing this risk. A shocking 87% of images include a high or critical vulnerability, up from the 75% we reported last year. When you view the data by number of vulnerabilities in images as opposed to number of vulnerable images, 71% of vulnerabilities have a fix available that has not been applied. Keep in mind, some images have more than one vulnerability. Organizations are aware of the danger, but struggle with the tension of addressing vulnerabilities while maintaining the fast pace of software releases.

87%

of images have high or critical vulnerabilities



13%

of images have low, medium or no vulnerabilities

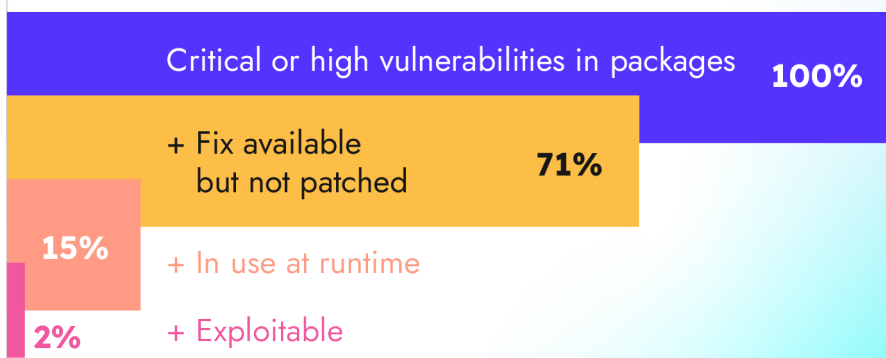
Fix what matters: 15% of high and critical vulnerabilities are in use at runtime

Although the list of software vulnerabilities to fix seems endless, there is an opportunity to reduce wasted time and improve the efficacy of cybersecurity programs. Our research found that only 15% of critical and high vulnerabilities with an available fix are in packages loaded at runtime. By filtering on those vulnerable packages that are actually in use, organizational teams can focus their efforts on a smaller fraction of the fixable vulnerabilities that represent true risk. This is a more actionable number that can allay some fears around release decisions and focus remediation efforts, provided organizations use the relevant security capabilities.

Runtime insights focus “shift-left” effort

Vulnerabilities are discovered in images every single day. However, it's not practical to fix every single one when you're maintaining multiple workloads at scale. Successful, modern vulnerability management requires security teams to prioritize vulnerabilities based on the actual or real risk to their organization.

Apply filters to prioritize vulnerabilities



There are a number of inputs commonly used to prioritize vulnerability remediation work, which include:

- [Common Vulnerability Scoring System](#) (CVSS) – specifies the severity of a known issue
- Exploitability – indicates if there is a known path for exploiting the vulnerability
- Fixable – identifies if there is a fix available to address the vulnerability

Addressing running, vulnerable packages with a known exploit should be the top priority. We found that our customers are proactive in fixing vulnerabilities that are exploitable and in packages loaded at runtime. When we combine multiple criteria of a vulnerability (fix availability, exploitability, and presence in a package loaded at runtime), what remains is 2% of the vulnerabilities found in the 25,000 images we analyzed.

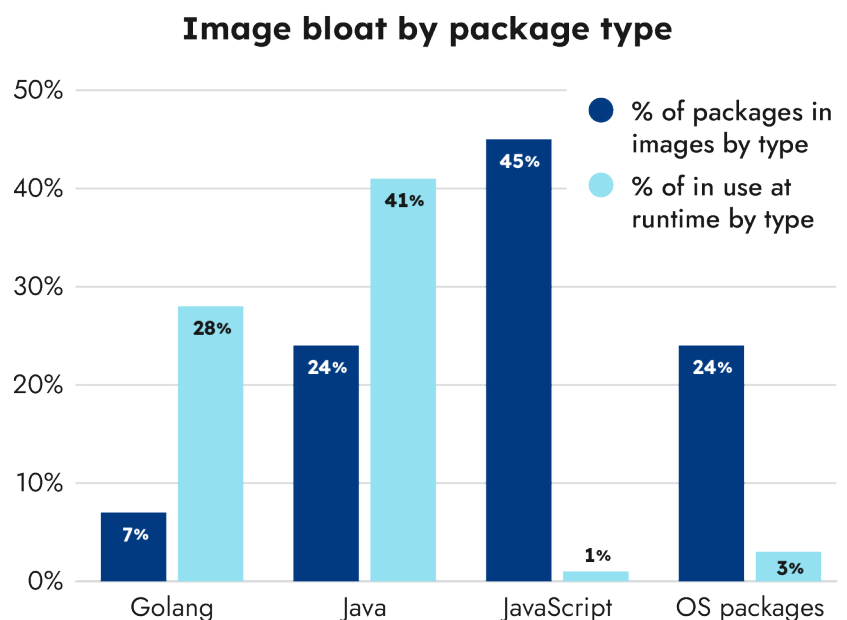
Addressing unpatchable vulnerability risk

Some vulnerabilities have an exploit available for attackers to use, but they do not have a readily available fix to mitigate potential threats. This small, but significant category affects security teams since they must still assess the risk of exploitable vulnerabilities and determine alternative mitigation strategies without Common Vulnerabilities and Exposures (CVE) patches or fixes.

When exploitable vulnerabilities must remain in your environment, one way security teams can ease the pain and reduce the risk of compromise is by implementing runtime security detections. Runtime protection is often powered by rules, but it should also employ a multi layered approach that incorporates behavior anomaly detection and AI or ML-based detection. This approach improves detection and mitigation of zero-day exploits and yet-unknown threats. Runtime protection mechanisms can also be tuned to detect novel threats that target vulnerable workloads in the unique environments of organizations. Detections can also be augmented with threat intelligence from threat research teams and regularly updated as new information or findings about behaviors become available.

Fewer than 1% of JavaScript packages are in use at runtime

Ideally, an image should only consist of the code necessary to do its job. Pre-packaged and open source images may include packages that are not required for your application. This is known as image bloat. Security teams can reduce their total number of vulnerabilities by removing the amount of unnecessary and unused packages that are often seen in third-party images.

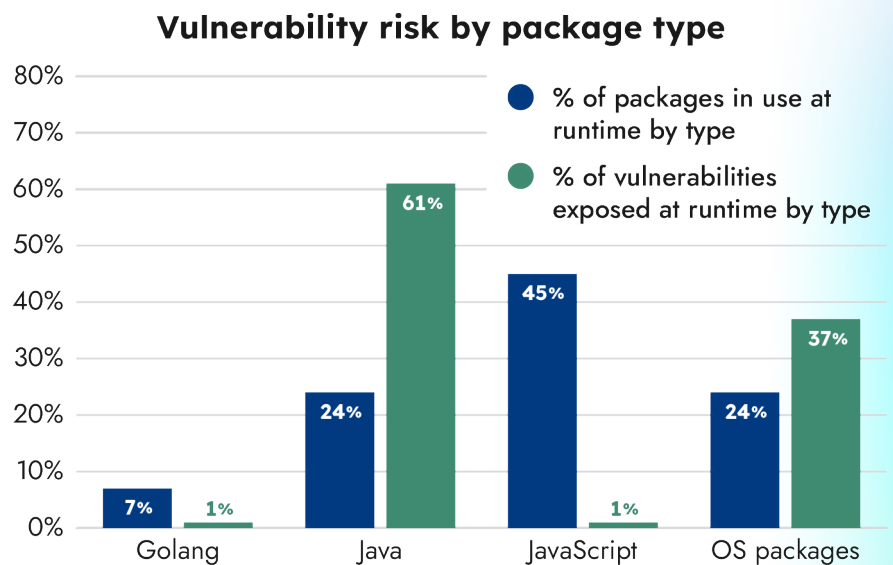


We looked at the package types of more than 6.3 million running images to determine the four most commonly used package types. Then, we analyzed those images to identify the types of packages that have the most bloat. JavaScript packages are found in the greatest number, yet fewer than 1% of them are loaded at runtime. This is a top candidate for removal to reduce bloat, and therefore, can also provide the greatest reduction in the number of vulnerabilities to fix.

Although it takes time to slim down images, doing so will reduce image scanning time and the number of vulnerabilities. Time spent streamlining packages and images ultimately results in time savings for delivery and runtime. Costs are amplified when you consider the cloud infrastructure costs and resources that need to be dedicated to running bloated workloads. To minimize image bloat, only include necessary packages, use an optimal base image, combine instructions and use multi-stage builds, and ensure you list the files you need in the COPY step.

Java packages are the riskiest, representing over 60% of vulnerabilities exposed at runtime

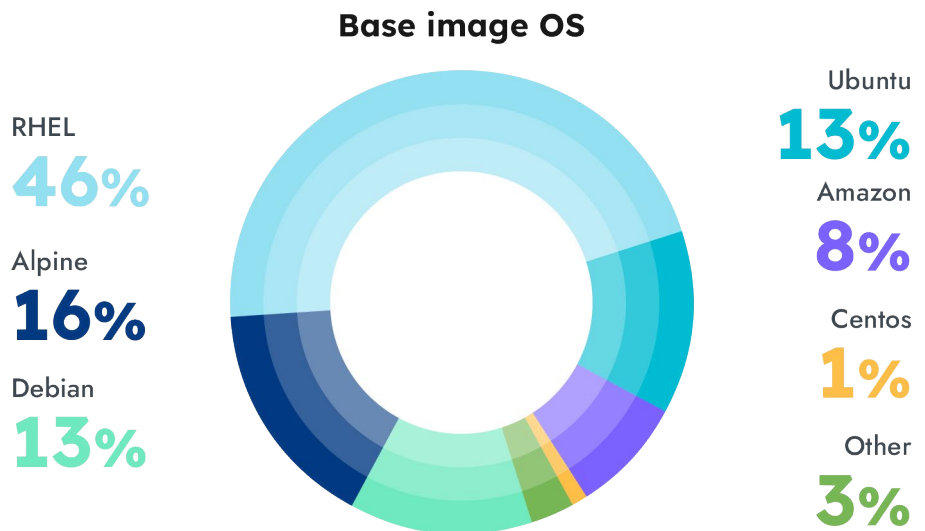
We measured the percent of vulnerabilities in packages loaded at runtime by package type to gauge which language, libraries, or file types presented the most vulnerability risk. Java packages were responsible for 61% of the more than 320,000 vulnerabilities in running packages. This was surprising, as Java packages only make up 24% of the packages loaded at runtime. Operating System (OS) packages were also risky, as they contained 37% of the vulnerabilities. Golang is a less common package type, but even after taking this into account, it has less vulnerability risk. Interestingly, while JavaScript packages are more prevalent, few of the packages are loaded at runtime, as shown in the chart to the right. They represent a significantly smaller percentage of vulnerabilities.



Base image OS selection can reduce bloat by 98%

Most people use a base image because it's easier than creating your own. Taking a look at our customer usage, we see that Red Hat Enterprise Linux (RHEL), which includes the Red Hat UBI (Universal Base Image), is by far the most popular at 46% of base images. This is up 10% year-over-year. This may be because RHEL has a long history of usage in the enterprise, and would be an easy choice as organizations move to cloud-native workloads. Interestingly, only 16% use Alpine, a lightweight Linux distribution.

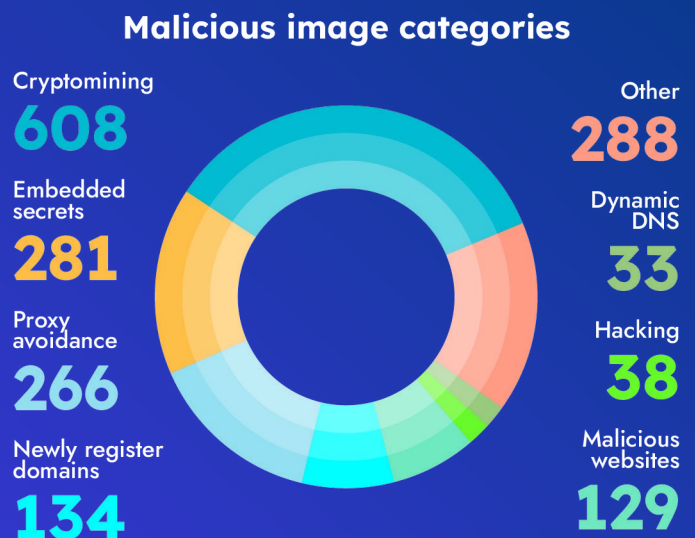
This is down from 25% last year. According to [crunchtools](#), the size of the standard uncompressed UBI image is 228MB and the Alpine image is 5.7MB. By using slimmed-down base images like Alpine, organizations can debloat their container environment by 97.5% and thereby reduce their attack surface. This will also reduce the number of OS vulnerabilities to fix, as only 8% of vulnerabilities are in OS packages loaded at runtime.



Dangerous images in public registries

In the [2022 Sysdig Cloud-Native Threat Report](#), the Sysdig Threat Research Team collected malicious images based on malicious IPs, or domains and secrets. Both pose a risk for users downloading and deploying publicly available images from Docker Hub, exposing their environments to attacks.

For the 1,777 malicious images identified, the chart indicates the type of nefarious content included in their layers.



03

Zero Trust: Lots of Talk, Little Action

Vulnerabilities are only one part of the cloud security story. Misconfigurations are still the biggest player in security incidents and, therefore, should be one of the greatest causes for concern in organizations. According to Gartner®, “By 2023, 75% of security failures will result from inadequate management of identities, access, and privileges, up from 50% in 2020.”^[1] Although many organizations are talking about zero trust principles, such as enforcing least privilege, our data shows little evidence of action.

90% of granted permissions are not used

It's a shocking finding we uncovered this year on the actual number of permissions used vs. the number of permissions given to users who are not administrators. The data showed that only 10% of permissions granted to non-admin users were utilized when analyzed over a 90-day window. We also found that admins are using a small fraction of their permissions. Of course, trimming permissions down to only what is required and minimizing the number of users with admin rights is critical to reducing risk.

It can be hard to know who needs what permissions in an organization. DevOps teams tend to grant more permissions than needed, so functionality works as expected and security is in the background. Furthermore, cloud vendors and their offerings are growing incredibly fast year-over-year. The continuous addition of services adds permissions too.

We often think of identity and permissioning in terms of humans, or traditional users. But applications, cloud services, commercial tools, and many other entities (or machine identities) must be authenticated and authorized appropriately as well. Similar to how applications on your cell phone request permissions to your contacts, photos, camera, microphone, and more, we must also consider access management for these non-human entities.



¹ Gartner, Best Practices for Optimizing IGA Access Certification, Gautham Mudra, 4 April 2022. Gartner is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.

58% of identities are non-human roles

Sysdig's year-over-year analysis implies that our customers are either granting access to more employees or maturing their Identity and Access Management (IAM) practices. The former scenario, a growth in human user population, may simply be a byproduct of moving more business into cloud environments or ramping up staffing due to business growth. Organizations may be maturing in their identity and access management practices by reducing the number of machine identities needed to run systems that must then be secured and maintained.

Last year, we found that 88% of identities in Sysdig customers' cloud environments were non-human roles. This year, that reduced to 58% non-human roles.

At the very least, organizations need visibility into all non-human identities and their relevant permissions. We found that more than 98% of permissions granted to non-human identities have not been used for at least 90 days. Oftentimes, these unused permissions are granted to orphaned identities,

such as expired test accounts or third-party accounts. Teams should apply [least privilege principles](#) to non-human identities in the same way they manage human identities. They should also remove unused test accounts wherever possible to prevent access risk. While this can be tedious to determine manually, in-use permission filters and automatically generated recommendations can make this process more efficient.

Cloud users and roles

Human identities
42%



Non-human identities
58%

“It’s critical for us to understand where we have overly permissive identities and due to the scale, we need an automated way to manage them. Trying to abide by the principle of least privilege, eliminating excessive permissions is a top security priority.”

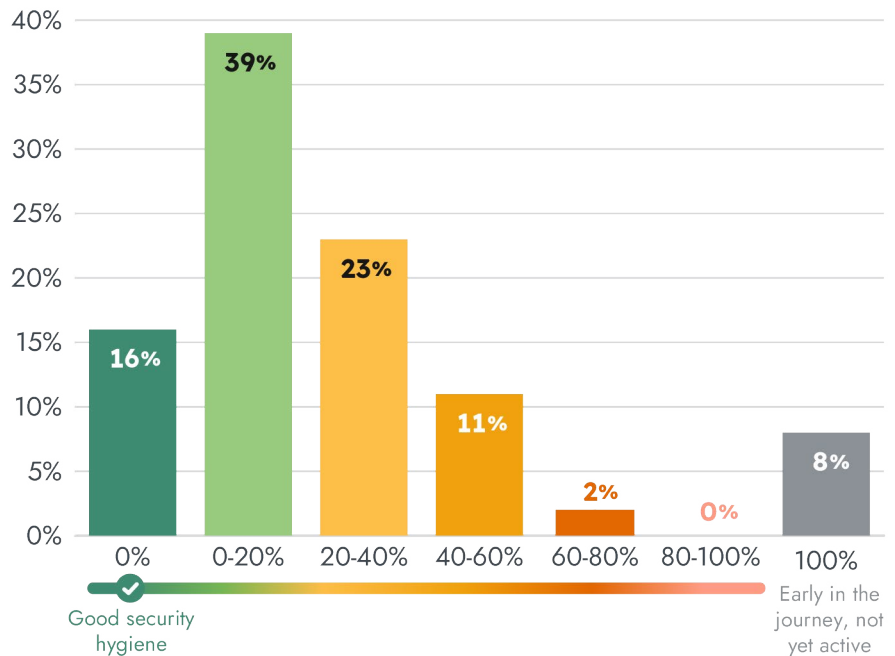
- **Georgia Bekiaridou,**
Security PO, Booking.com

16% of users have strong best practices for accounts

Last year, we saw that 27% of Sysdig customers were using their cloud environment's root user account for administrative and daily tasks. [Cloud security best practices](#) and the CIS Benchmark for AWS indicate that organizations should avoid doing so and suggest creating dedicated roles with limited, yet appropriate permissions for performing administrative tasks.

This year, we took a closer look at all of Sysdig's customer accounts with administrator permissions and decided to calculate a risk score. The risk scores consider the percentage of a customer's cloud accounts with poor security hygiene. We defined this as an account with administrator access, no [multi-factor authentication](#) (MFA) enabled, and account inactivity of 90+ days. These are conveniences attackers look for because they provide easier account access and reduced chance of detection.

IAM vulnerability hygiene rating



04

Mature Organizations Are Proactively Testing Their Security Posture

The prevalence of unpatched vulnerabilities, overprovisioned identities, and risky configurations highlights the need to detect anomalous behavior and immediately investigate potential threats.

Top MITRE tactics detected: Defense evasion and privilege escalation

Based on the analysis of our [Falco rules against the MITRE ATT&CK® Framework](#), we saw evidence that rules labeled for privilege escalation and defense evasion are most often triggered. Fortunately, most of these events did not indicate actual malicious attacks, but rather are attributed to customers using proactive threat analysis to understand where to implement security guardrails. We saw three causes for these rules to be triggered. First, there is evidence that security scans were a leading cause, an indication that teams are proactively testing the ability to detect and block intrusion attempts.

Room to improve testing procedures to reduce alert fatigue

Next, some events were triggered by services that legitimately require escalated privileges. These alerts can be addressed by tuning rules appropriately for specific services. Lastly, we believe some percentage of the alerts are a result of poor practices, which can be addressed by defining and following [container security best practices](#). For example, package management should not be done in live containers. Package managers download tools, run binaries, and make a lot of file system changes that can trigger alerts.

Triggered rules based on MITRE ATT&CK® Framework tactics

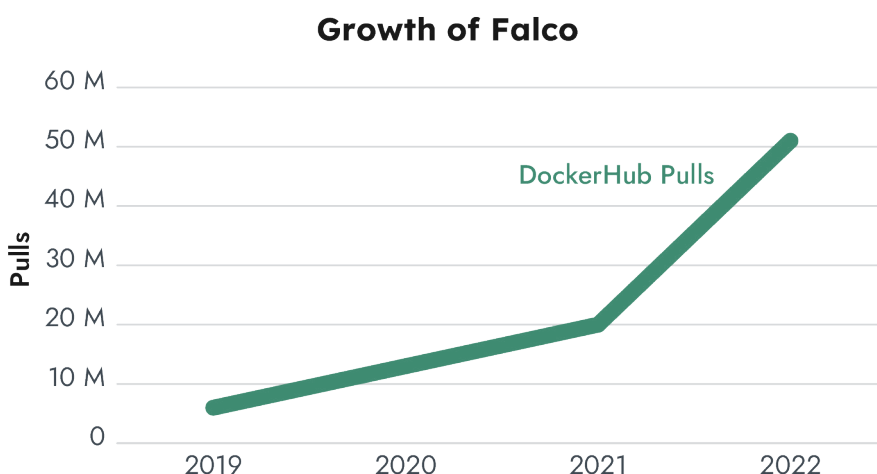


The Sysdig Threat Research Team (Sysdig TRT) builds Falco rules based on automated threat feeds, manual analysis of open source, and data gathered from their managed honeynet. Any detection rules should be regularly tuned by your vendor to adjust to the ever-changing threat landscape, and be personally tuned based upon your own activity that may cause false positives.

For example, the Sysdig TRT regularly updates the out-of-the-box rules *Outbound Connection to C2 Servers* and *Malicious Filenames Written* with known and new nefarious activity. In addition, we see our customers focus on capturing log activity related to persistence and privilege escalation rules. A small number of our customers are even modifying and customizing out-of-the-box Falco rules provided by Sysdig, an indication of their security maturity as they improve their detections and reduce false positives.

Major ecosystem vendors are adopting Falco








The presence of unpatched vulnerabilities and overprivileged identities highlights the need for runtime security. Container runtime threat detection is going mainstream with large ecosystem providers recognizing the need to help customers address security for cloud-native applications. The major cloud providers, along with many key technology vendors, are taking advantage of an open standard, building and recommending solutions based on Falco.



Falco, the open source project created by Sysdig

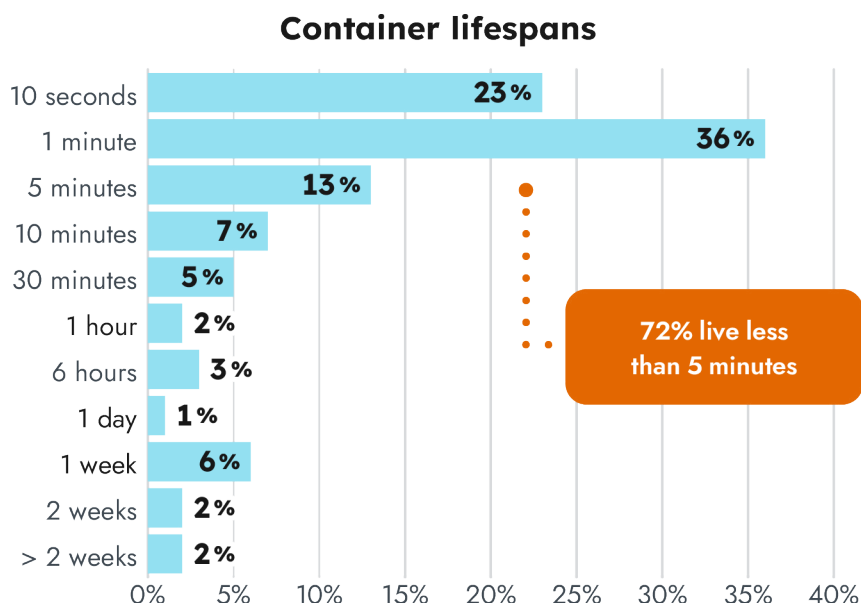
Falco provides real-time visibility into your containers, hosts, and cloud services, detecting unexpected behavior, intrusions, and data theft. Falco was created by Sysdig and contributed to the Cloud Native Computing Foundation (CNCF).

Broad adoption of Falco detection rules and libraries

	AWS Fargate serverless runtime security
	gVisor container sandbox security
	Microsoft Defender data collection
	HPE Ezmeral runtime security
	SysFlow telemetry and security
	Data collection
	Anomaly detection

72% of containers live fewer than five minutes

The container lifespan was already short, with nearly half of containers living fewer than five minutes in past years. However, this year, the number increased to more than 70% of containers living less than five minutes. This is a big jump and reinforces the need for continuous threat detection and to capture a record for investigations, as the container may live for only seconds. There is no way for us to know for sure, but we speculate that companies are becoming more efficient as they mature and are using containers to run more short-lived functions similar to how you might use a serverless environment.



Why do containers have such a short life? Many containers only need to live long enough to execute a function and then terminate when it's complete. Seconds may seem short, but for some processes, it's all that is required. The ephemeral nature of containers remains one of the technology's unique advantages, in that container images are designed to change as needed. However, it also presents issues to consider for monitoring, security, and compliance because many tools can't report on entities that no longer exist.

83% of containers run as root

As organizations focus on fixing vulnerabilities, they may not be scanning for common configuration mistakes. We saw an increase from 76% to 83% of images running as root, allowing for privileged containers to potentially be compromised.

From talking to our customers, in practice, even if risky configurations are detected at runtime, teams do not stop these containers as they do not want to slow their deployment. Instead, they run within a grace period and then decide on the remediation step. Although some containers require this level of privilege to perform their intended function, this number is shockingly high and the trend is going in the wrong direction.

83%
of containers
run as root

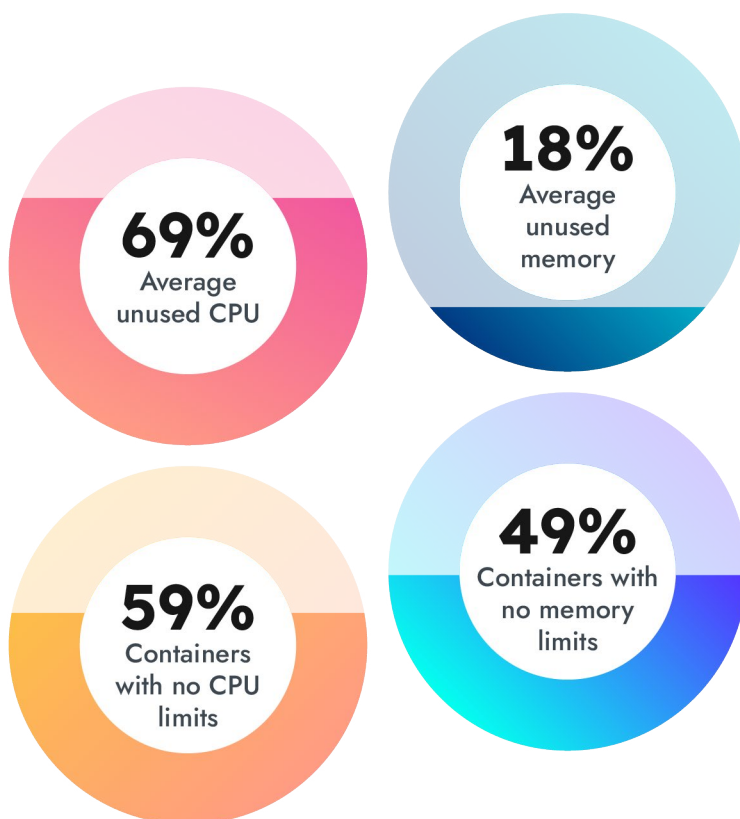
05

Millions Wasted on Unused Kubernetes Resources

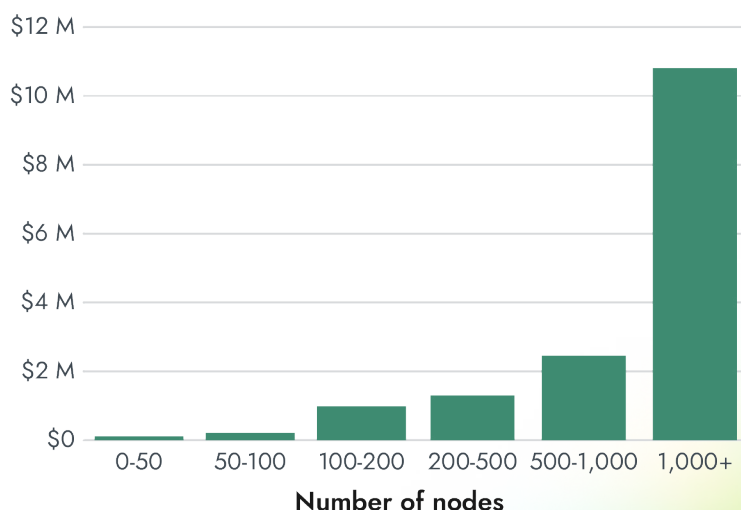
In an ephemeral, dynamic environment like Kubernetes, keeping track of cost and usage is inherently difficult. Organizations often neglect to set limits on how many resources a container can use. In addition, environments where developers are allowed to choose their own capacity needs can lead to overallocation and these are rarely audited and right sized. In looking across the customers in our largest region, we found that 59% of containers had no CPU limits defined and 49% had no memory limits defined. In terms of unused resources, an average of 69% of requested CPU cores and 18% of requested memory were unused.

Digging deeper into container efficiency, our internal reports indicate that on average, 69% of containers are using fewer than 25% of requested CPU resources.

Without knowing the utilization of their clusters, organizations could be wasting money due to overallocation or causing performance issues by running out of resources. Given the average cost of AWS pricing, organizations with around 150 Kubernetes nodes could be spending up to \$980,000 per year more than they need to due to underutilized CPU resources. Companies with larger deployments, between 200 and 500 nodes, could be spending up to \$1.3 million per year for unused resources, while organizations with more than 1,000 nodes could reduce their wasted spending up to \$10.8 million per year.



Cost savings per year



06

Usage Trends and Insights

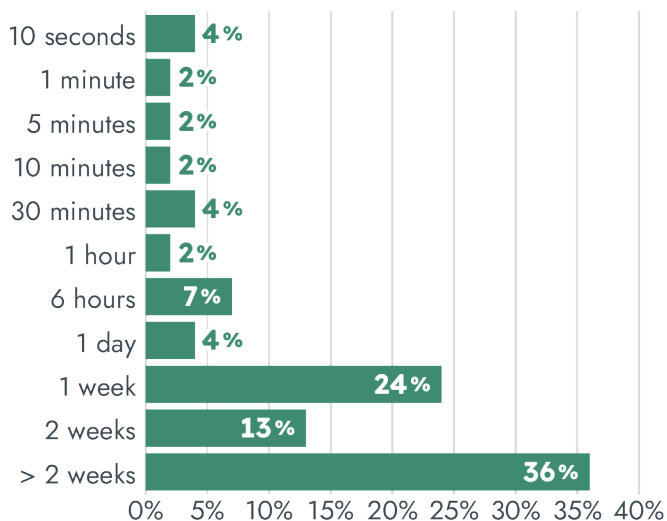
Continuous development and image lifespans

Containers are a perfect companion to the agile movement, accelerating the development and release of code, often as containerized microservices. Our image lifespan data reflects the shift in the time between code releases and the reality that CI/CD pipelines are helping developer teams deliver software updates at a faster cadence than ever before. The data shows that about half of container images get replaced, also known as churn, in a week or fewer. For most, if not all, of today's businesses, speed to market matters and makes all the difference in maintaining competitiveness. Code is being deployed more frequently, which creates new container images. Containers give businesses what they need to turn great ideas into reality, fast.

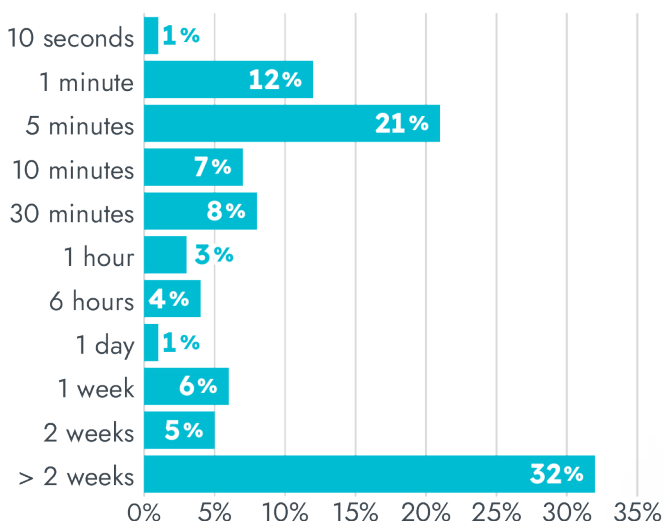
Service lifespan

Services — the functional software components of our applications like database software, load balancers, and custom code — are continuously being improved. However, at the same time, it's important to keep services up and running around the clock to be able to meet customer expectations. The data show that service lifespans have remained relatively consistent compared to last year.

Container image lifespans



Service lifespans

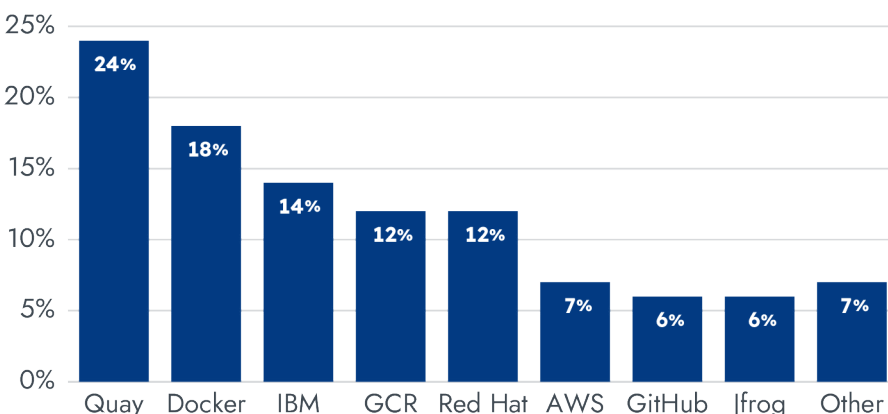


Container registries

Container registries provide repositories for hosting and managing container images. This year, we saw the adoption of Red Hat and IBM registries both double in usage. Quay and Docker use has reduced slightly to a combined 42% of customer adoption.

We also compared the number of containers pulled from public and private repositories. We found that the split between public and private sources changed slightly in comparison to last year, with public source trust reducing from 61% to 56%. Using public registries poses a risk because few are validated or checked for vulnerabilities. In some cases, the convenience of using public repositories may outweigh the risk, but the best practice is to enforce explicit policies about which registries are approved for use in the organization. This year-over-year increase in the use of private registries indicates security maturity as organizations shift away from public registries.

Container registries



Images pulled from registries



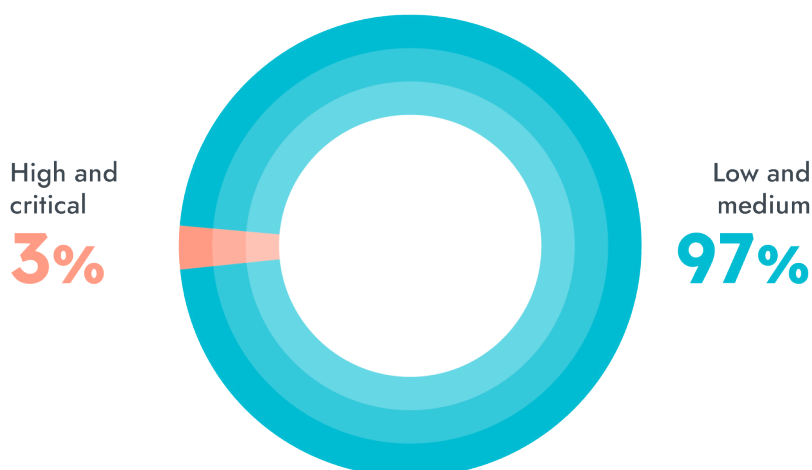
Image scanning

Whether the container images originate from private or public registries, it is critical to scan them and identify known Common Vulnerabilities and Exposures (CVE) prior to deploying into production. We assessed all the images our customers deployed for OS and non-operating system vulnerabilities. We found that OS packages have fewer flaws than non-OS packages, likely due to the fact that they are usually supported and maintained by industry vendors.

OS vulnerability snapshot

We noticed that 3% of OS vulnerabilities are high or critical, relatively unchanged from last year. Although this may seem low, if an OS vulnerability is exploited, it can compromise your entire image and bring down your applications. Additionally, OS vulnerabilities can have a very large blast radius because many different workloads are affected at the same time.

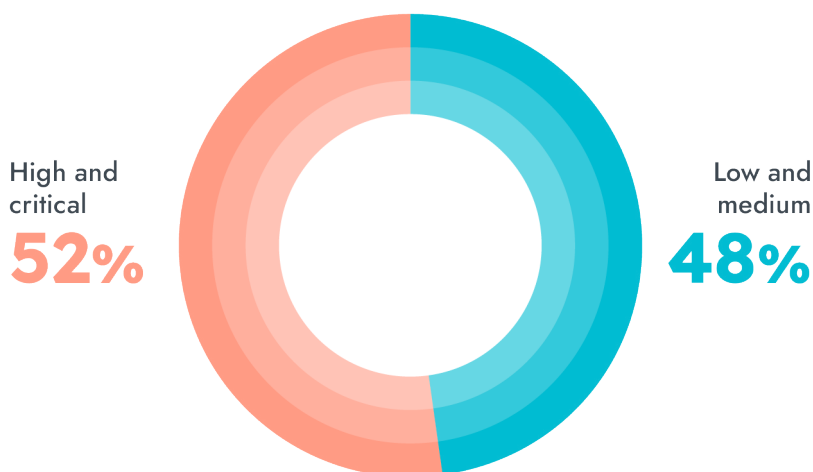
OS vulnerabilities by severity



Non-OS vulnerability snapshot

What many teams don't check for are vulnerabilities in third-party libraries, likely under the presumption that a package released by someone else has been secured and is regularly maintained. We found that 52% of non-OS packages have high or critical severity vulnerabilities, only a slight reduction from last year. Developers might be unknowingly pulling in vulnerabilities from non-OS open source packages, like Python PIP or Ruby Gem, and introducing security risk. Gaining visibility into third-party dependencies and determining whether they are truly exploitable has always been a challenge, but runtime context enrichment can enable actionable prioritization of these types of flaws.

Non-OS vulnerabilities by severity



Scanning in build phase vs. runtime

There is no indication of security program maturity regarding where testing takes place in the development lifecycle. In comparing where images are scanned in the workflow for the first time, the numbers are relatively unchanged over the last year.

A likely reason scanning in runtime is still so high is due to the use of third-party software downloads from vendors. These are typically considered trusted sources, so DevOps teams may presume the images are secure and save the time and effort of scanning in the CI/CD pipeline, skipping to scanning in runtime. However, as a friendly reminder in the spirit of "shifting left," it is best to scan images in the CI/CD pipeline to ensure security prior to deployment. The slight year-over-year decrease in runtime scanning tells us that either the "shift left" is starting to happen, or organizations are maturing and moving from vendor-provided images to custom-built images.

Where images are scanned



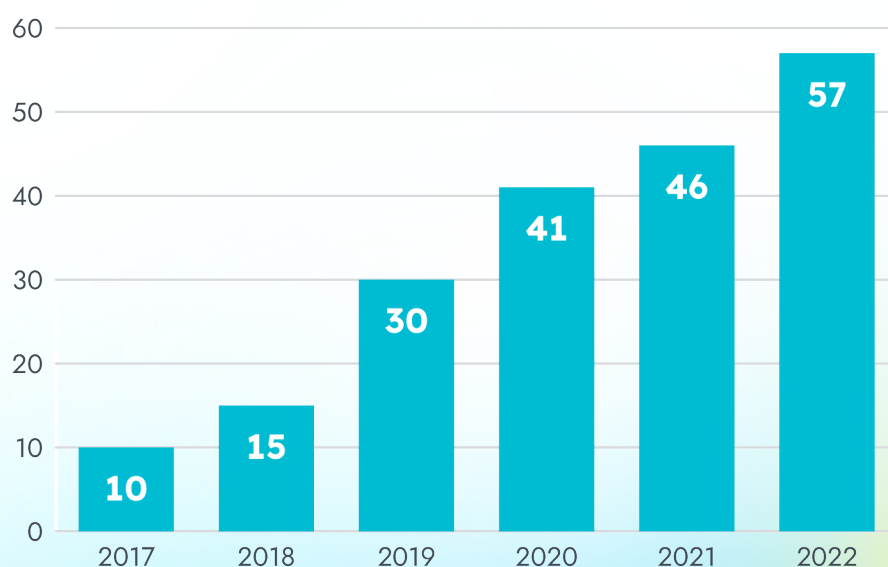
Containers and Kubernetes

Each year, we take a look at details specific to the count and activity around containers and Kubernetes, including density and lifespans. This provides insight into the rate of adoption, but also illustrates the scale and efficiencies being achieved. In this section, we also answer questions like: How many clusters are customers operating? How many pods run per node? How much capacity does a cluster use? We look at a range of details about what customers are doing with Kubernetes. Because Sysdig automatically collects Kubernetes labels and metadata, we're able to provide cloud-native context for all of the data insights we discover, from performance metrics and alerts to security events. This same capability enables us to capture each of the following usage metrics from the cluster all the way to pods and containers, all with a simple query.

Container density

Over the past six years, the median number of containers per host increased in every report. This year, that number jumped again by 24% year-over-year to an average of 57. It is possible that organizations are learning how to be more efficient by either using larger instance sizes or smaller containers. While the primary goal of containers is to speed development and deployment, many organizations are benefiting from increased utilization of hardware resources thanks to container efficiencies.

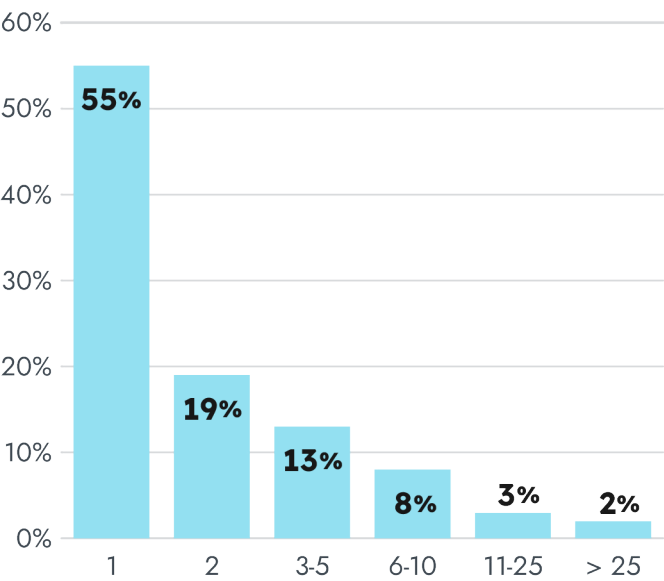
Median containers per host



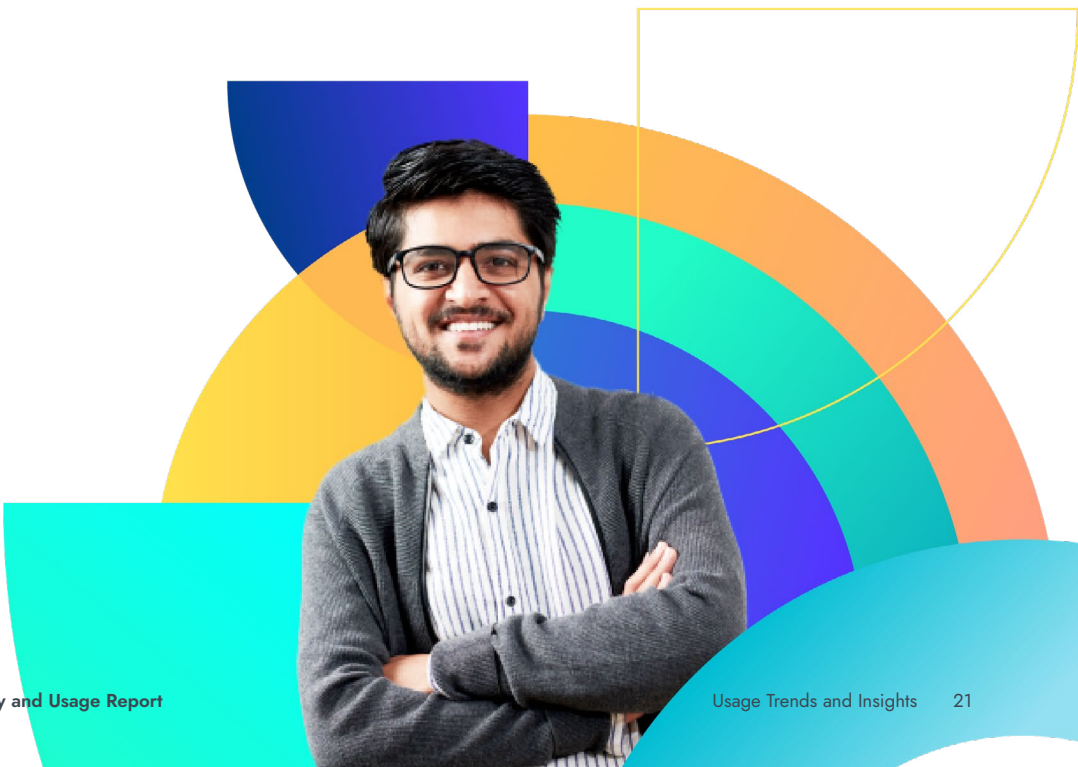
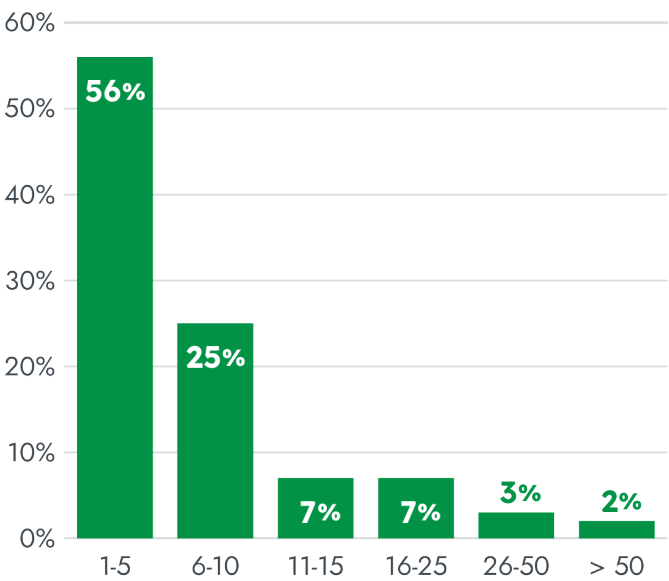
Kubernetes clusters and nodes

Some customers maintain a few large clusters, while others have many clusters of varying sizes. The charts in this section show a distribution of cluster count and nodes per cluster for users of the Sysdig platform. The large number of single clusters per customer, and relatively small number of nodes, is an indication that many enterprises are still early in their use of Kubernetes. We've also recognized that the use of managed Kubernetes services in public clouds is another factor that impacts these data points. This year, we observed a shift towards fewer large clusters overall and more nodes per cluster. This may indicate that cloud-native deployments are starting to mature by utilizing resources more efficiently.

Number of clusters



Number of nodes per cluster



Kubernetes namespaces, deployments, and pods

Namespaces

Kubernetes namespaces provide logical isolation to help organize cluster resources between multiple users, teams, or applications. Kubernetes starts with three initial namespaces: default, kube-system, and kubepublic. How namespaces are used varies across organizations, but it is common for cloud teams to use a unique namespace per application.

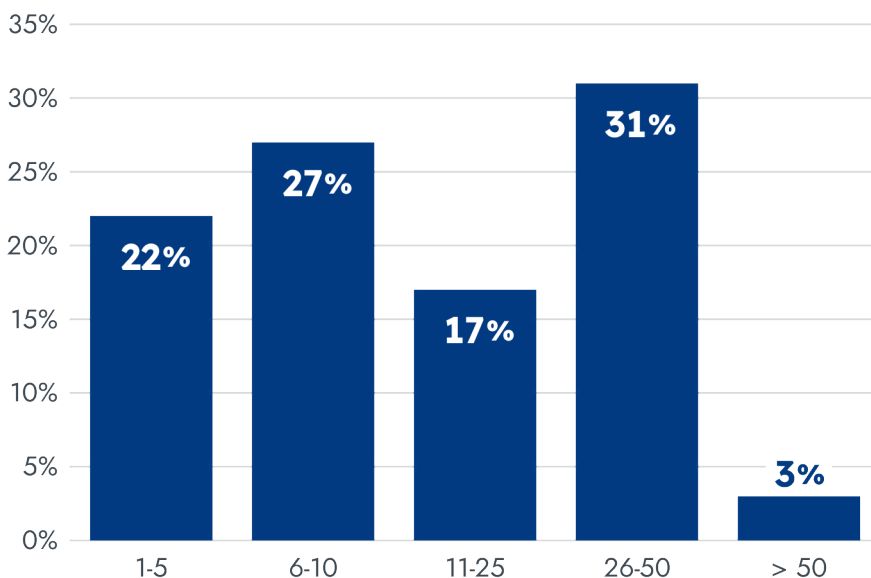
Deployments per namespace

Deployments describe the desired state for pods and ReplicaSets, and help ensure that one or more instances of your application are available to serve user requests.

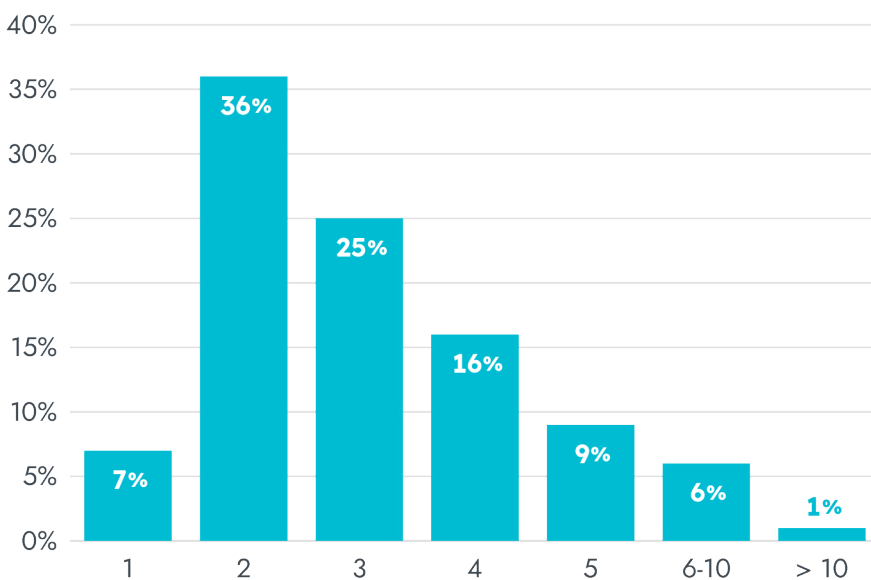
Deployments represent a set of multiple, identical pods with no unique identities, such as deployments of NGINX, Redis, or Tomcat. The number of Deployments per namespace provides an idea of how many services compose our users' microservices applications.

We saw a slight shift this year toward more namespaces per cluster and more deployments per namespace. Again, this may indicate a maturing of these cloud-native environments.

Namespaces per cluster



Deployments per namespace



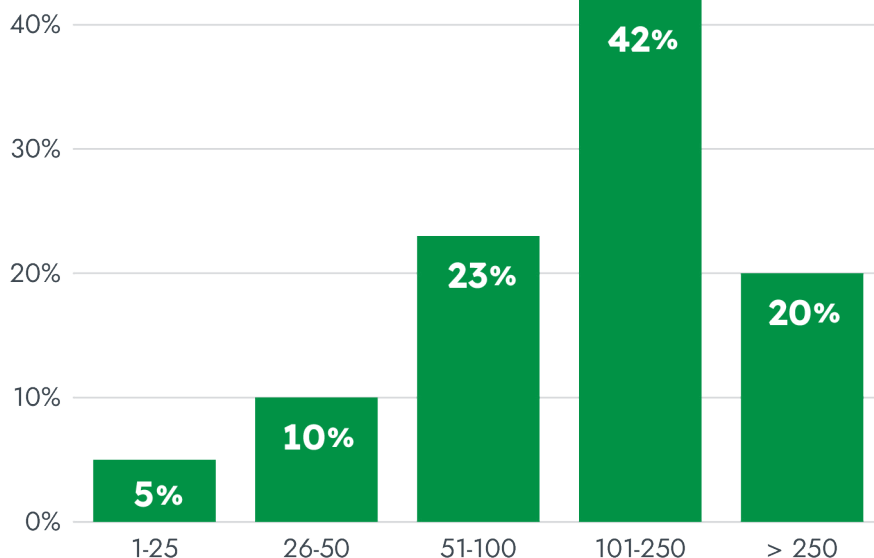
Pods

Pods are the smallest deployable object in Kubernetes. They contain one or more containers with shared storage and network, as well as a specification for how to run the containers.

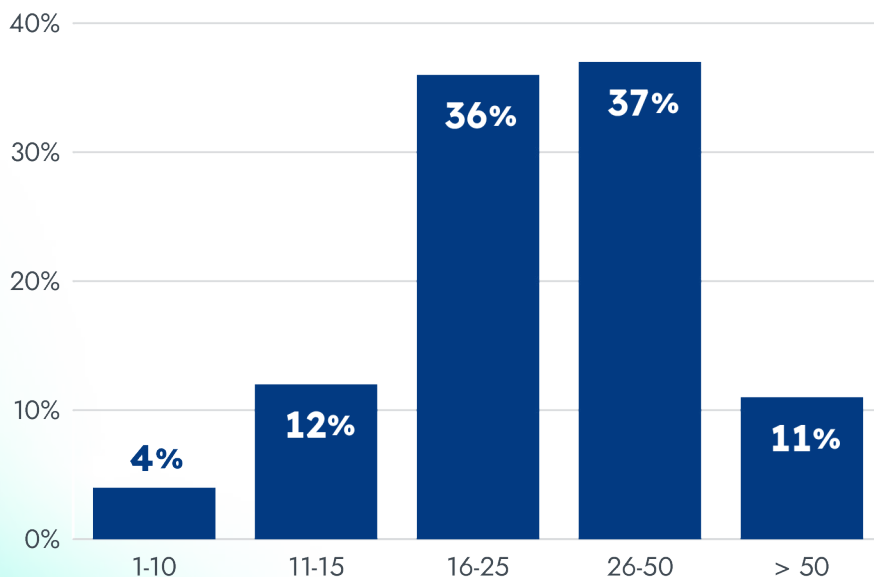
Pods per node

A pod remains on a node until its process is complete, the pod is deleted, the pod is evicted from the node due to lack of resources, or the node fails. This year, we saw a significant increase in the number of pods per cluster, with 54% of our customers running more than 100 pods. That number increased again to 62% this year. Similarly, the number of customers running more than 25 pods per node also increased from 28% last year to 48% this year, indicating that customers are running fewer clusters (as seen above) with more pods deployed on those clusters.

Pods per cluster



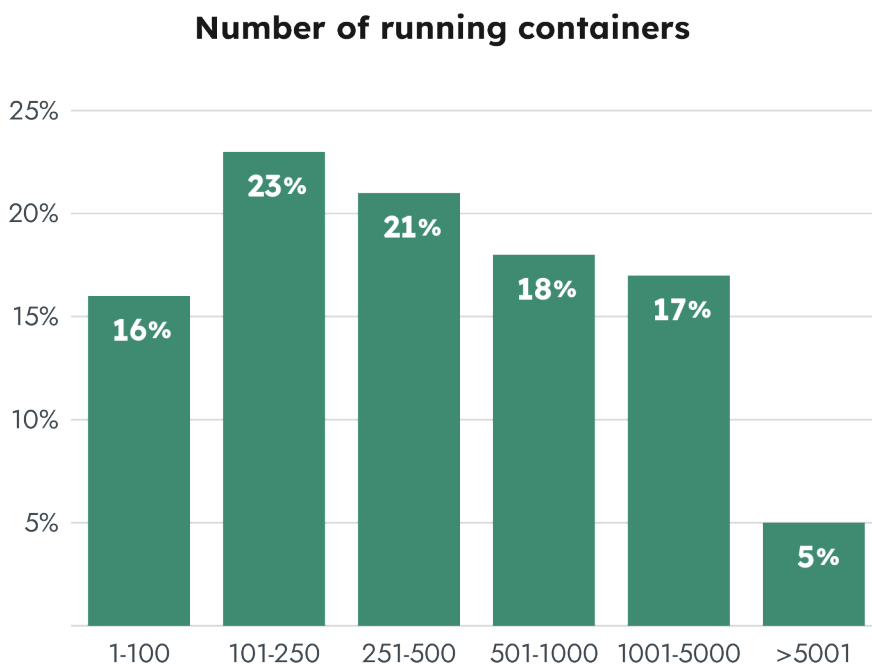
Pods per node



Containers, images, and alerts

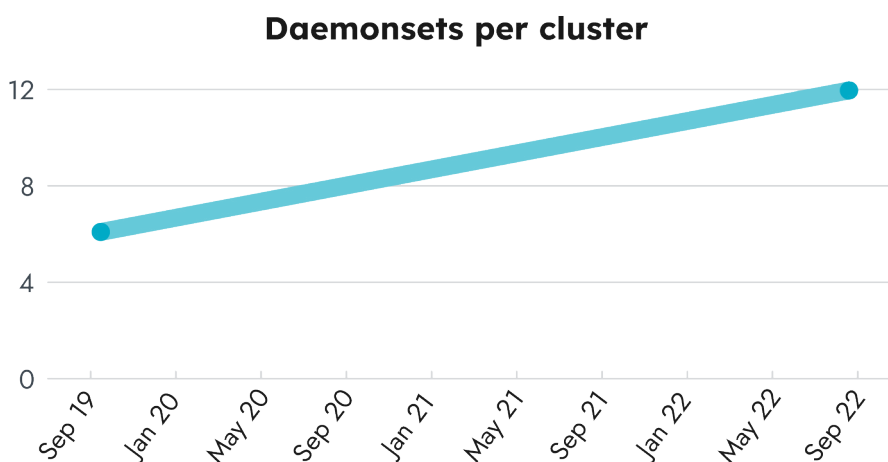
Containers-per-organization

To get a sense of the scale at which enterprises are currently operating, we looked at the number of containers each customer runs across their infrastructure. 61% of customers run more than 250 containers. At the high end, only 6% of customers are managing more than 5,000 containers. DevOps and cloud teams report that once the benefits are proven, adoption accelerates as more business units look to onboard to the new platform. However, this year showed movement toward an overall increase in the number of running containers. This shift may indicate that either more workloads are moving to containers and away from traditional architectures, or that the infrastructure is increasingly efficient and able to handle the growing number of containers.



Daemonsets per cluster show steady growth

Daemonsets can be deployed once and ensure a service is running on every node in a Kubernetes cluster. This makes for a simple way to deploy services that need to be running everywhere without having to deal with individual configurations. However, a service deployed as a daemonset that is using too many resources can lead to widespread performance issues. It is critical to keep an eye on your daemonsets and make sure that they are performing properly. We have seen an increase in the use of daemonsets growing from an average of six per cluster in January 2020, to an average of 12 per cluster in January 2023.

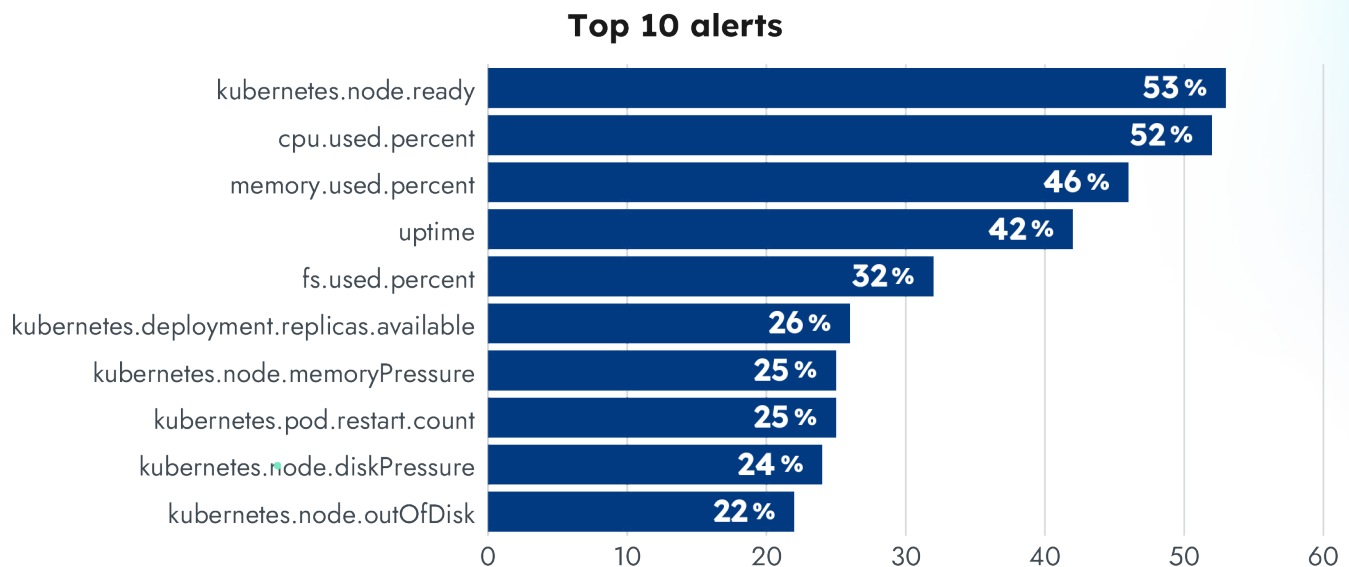


Alerts

Analysis of trends with the types of alerts set by our customers helps us understand the kind of conditions that our users identify as having the most potential for disruption to their container operations.

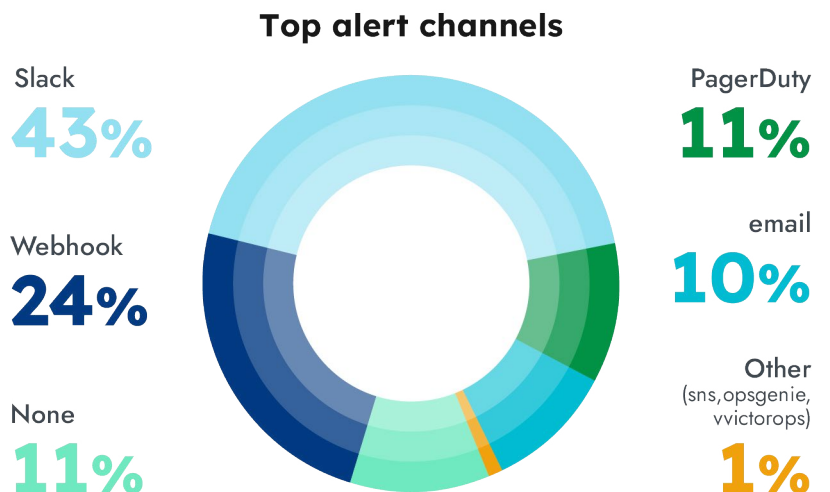
The top 10 alert conditions

There are more than 800 unique alert conditions being used across our customers today. The graphic below represents the most commonly used alert conditions, along with the percentage of customers using each. Kubernetes.node.ready continues to be the most common, along with important resource and uptime metrics. These most used alerts have remained consistent over the past three years.



Alert channels

We looked at the communication channels users have configured to receive alerts. The use of Slack increased this year, growing from 36% a year ago to 43%. It's likely that Slack is being used for non-critical alerts handled during normal work hours, while solutions like PagerDuty are being used for "waking people from bed." The shift to remote work could have played a role here, as Slack usage in general has increased due to this growing trend. The use of webhooks also grew from 14% last year to 24% this year. As work environments change, new tools are likely to be adopted which would increase the use of webhooks when there are no integrations already built for those new tools.



There are a number of alerts that don't have a notification channel configured, but this isn't necessarily a bad thing. This could be because the alert was for informational purposes only, or because the Sysdig platform itself provided enough information to satisfy the demands of the alert in question.

What Services are Customers Running?

The top open source solutions running in containers

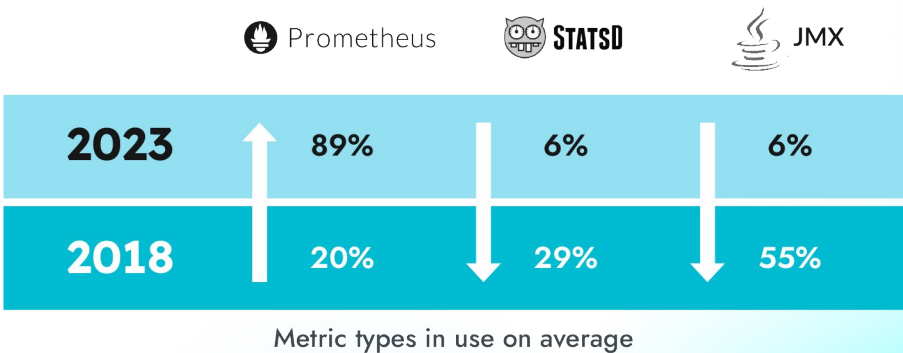
Open source software has changed the face of enterprise computing. It powers innovation across not just infrastructure, but especially application development. Sysdig's ability to auto-discover the processes inside containers gives us instant insight into the solutions that make up the cloud-native services that our customers run in production. Below are the top 12 open source technologies deployed by Sysdig customers:

Given the wide range of options available in the open source community, it's surprising that the most used services in our list have remained fairly consistent over the past four years. This year, we saw that NGINX, Go (aka golang), and Java all had significant increases. This is most likely due to an increase in cloud-native development, since these services are often used when building out applications in the cloud. We purposely omitted Kubernetes components like etcd and fluentd, as well as Falco. Because these are deployed by default, they end up at the top of the list for every Kubernetes user.



Custom metrics

Custom metric solutions give developers and DevOps teams a way to instrument code to collect unique metrics. This approach has become a popular way to monitor applications in production cloud environments, along with tracing and log analysis. Of the three mainstay solutions, JMX, StatsD, and Prometheus, Prometheus maintained a strong lead with slight growth to 89% of all custom metrics collected. StatsD fell by half, from 13% to only 6%, while JMX metrics stayed about the same. As the use of new programming frameworks expands, alternatives like JMX metrics (for Java apps) and StatsD continue to decline. It is clear that with the strong connection between Prometheus and Kubernetes, more organizations are adopting Prometheus metrics as they move toward cloud-native architectures.

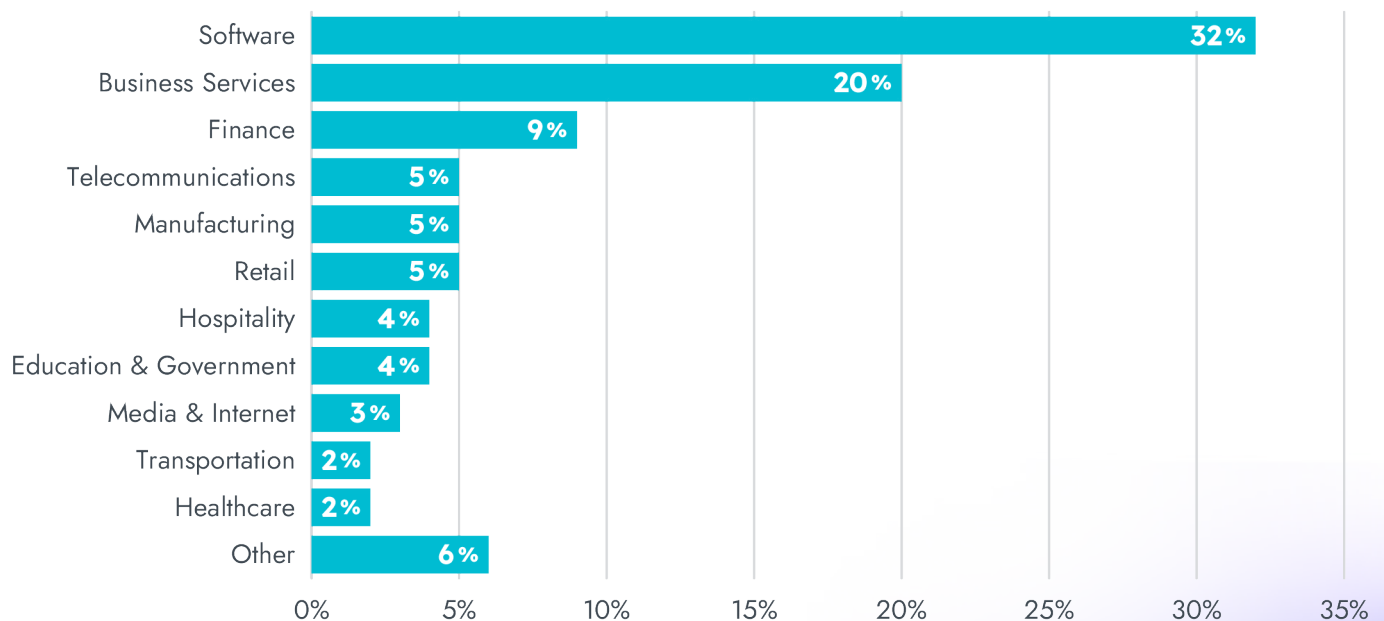


07

Methodology

The data in this report is derived from an analysis of more than seven million containers that our customers are running on a daily basis. We also pulled from public data sources like GitHub, Docker Hub, and the CNCF. The data originates from container deployments across a wide range of industries with organizations ranging in size from mid-market to large enterprise. Anonymized customer data was analyzed across North and South America, Australia, the EU, UK, and Japan.

Industries



08

Conclusion

Companies are rapidly adopting containerized microservices, CI/CD, and on-demand cloud services to speed innovation. However, the pace of change opens the door to risk as cloud sprawl and the complexity of cloud-native applications expose a lack of maturity in DevSecOps processes. In addition, supply chain risk from misconfigurations and vulnerabilities has emerged as a major area of concern.

Our research demonstrates that although there is awareness of required tools and the benefits of zero trust approaches, cloud security processes still lag behind the fast pace of cloud adoption. From the real-world customer data we examined, there are several security practice areas that require improvement to reduce risk:

- **Identity and access management:** The large disparity between permissions granted vs. those required highlights the urgent need to regularly measure and manage permissions to reduce opportunities for attack.
- **Vulnerability management:** With a majority of container images running with risky vulnerabilities in production, teams must address image bloat and focus their remediation efforts by prioritizing vulnerabilities based on real runtime risk.
- **Detection and response:** Privilege escalation and defense evasion attacks are top of the threat list for our customers. To stay ahead of the evolving threat landscape, threat detection rules should be regularly updated to spot nefarious activity.

Beyond security, this year's data demonstrates the opportunity for organizations to reduce cloud costs by addressing unused Kubernetes resources. Time invested in capacity planning can yield a strong return. By implementing proper container resource limits and continuous monitoring, organizations will be able to keep costs in check without risking application performance.

The key trends from our sixth annual report highlight the continued growth in container environments, and the growing dependency on open source-based solutions to run and secure them. The market for automated and scalable tools designed for the cloud and containers continues to expand, helping teams more effectively detect threats and risks with no blind spots, focus on the actions with the greatest impact, and avoid wasting time.

Thank you for reading the Sysdig 2023 Cloud-Native Security and Usage Report. We look forward to following and documenting the evolution of the container market in the coming year. See you then!

Additional Resources

Sysdig Secure delivers cloud and container security so you can stop breaches with no wasted time. We created open source Falco, the standard for cloud-native threat detection. Using Falco, the platform provides real-time threat detection with the detail to immediately respond. With Sysdig you can prioritize vulnerabilities, trim excess permissions, and fix misconfigurations based on in-use risk exposure. Manage cloud costs and rapidly troubleshoot issues using our observability offering. The largest and most innovative companies around the world rely on Sysdig for cloud and container security.

Want to dig deeper? Check out the [Sysdig blog](#) to stay on top of cloud-native news and industry best practices.

CHECK OUT OUR BLOG

