sysdig

2019 Container Usage Report

Five minute container life highlights need for specific security controls

REPORT

Contents

Executive Summary

The past year has seen continued momentum for Kubernetes as the dominant enabler of container-based applications. As more enterprises adapt to cloud-native architectures and embark on multi-cloud strategies, demands are changing not just usage patterns, but processes and organizational structures as well.

It's well known that containers are ephemeral. What's surprising is that over half of containers are alive for less than five minutes. As a result, organizations have recognized that security tools and processes have to be different. Cloud teams are integrating specific security and compliance checks into their DevOps processes to better understand and manage risk.

For the past three years, we've provided insights into container usage through real-time, real-world customer data. This data represents usage at companies around the world, from a broad range of industries. Our unique vantage point lets us discover details about the current use of infrastructure, applications, and containers, as well as security and compliance. Armed with these insights, we bring you the Sysdig 2019 Container Usage Report.

Key 2019 Insights

52% of containers live 5 minutes or less

2x the number of containers alive for 10 seconds or less

100% increase in container density year over year

GO — Go and Node.js overtake Java as top cloud app frameworks — node JS

Prometheus rises to lead custom metric solution

Red Hat OpenShift is **top choice** for secure, on-prem Kubernetes

Containers frequently run as **root and in privileged mode**

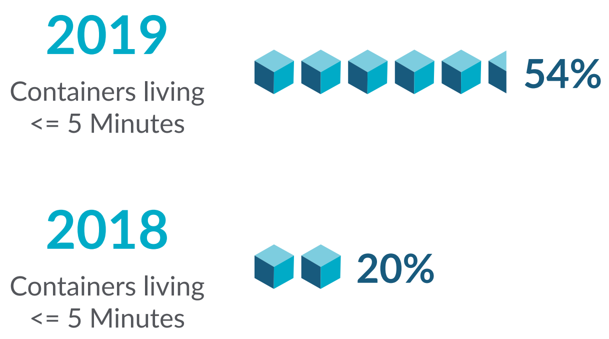
2019 Container Usage Report

This year, we've incorporated additional data sources to explore new data points and dig deep into Kubernetes usage patterns. For the first time, we showcase the security and compliance concerns and issues faced by our customers.

We believe providing visibility into enterprise use of containers and surrounding technology helps cloud teams, and the industry as a whole, understand trends and identify opportunities for operating Kubernetes and containers in production.

The findings in the following pages are a snapshot of enterprise usage across well over two million deployed containers that are running in production and are secured and monitored by Sysdig software. For the first time, we've incorporated usage data from customers who deploy the Sysdig Secure DevOps Platform in private data centers — many of whom operate some of the largest container deployments in the world. Also this year, we've taken a snapshot of usage from the Sysdig service offered in IBM Cloud since December 2018. This data, combined with our own SaaS cloud offering, provides a broad spectrum of detail cross an extensive set of customers.

Container Lifespan 2019 vs. 2018



> **" Short-lived containers are a big security challenge. Processes start and stop so quickly that it's easy to miss suspicious activity."**
>
> –
>
> Head of Security and Compliance
> SaaS Software Company

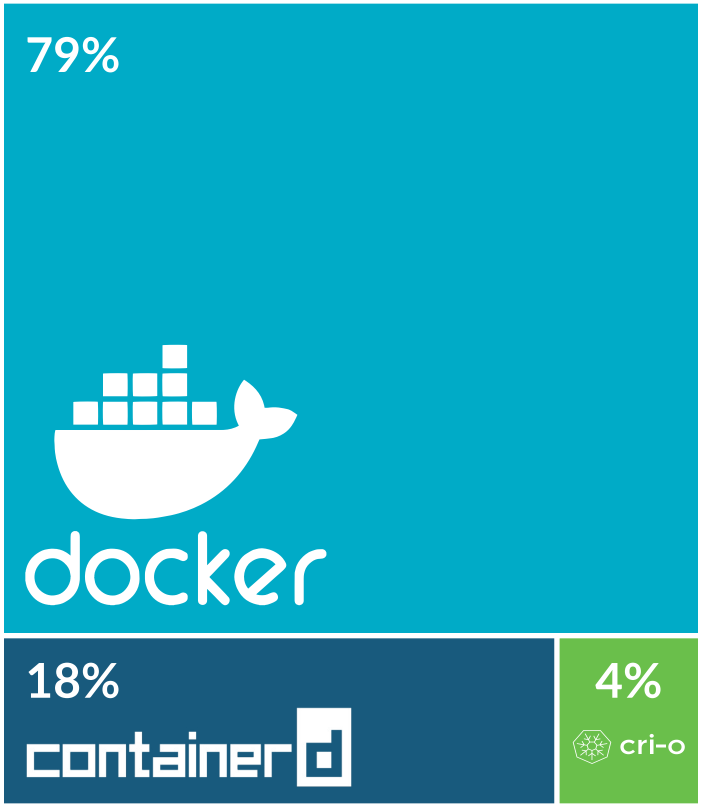
What Container Platforms are Being Deployed?

Container runtimes

In our 2018 report, we described how the Open Container Initiative (OCI), the Linux Foundation project focused on designing open standards for operating-system-level virtualization, was helping usher in alternate container runtimes. This has happened in a big way in 2019, with containerd grabbing a significant share. To be fair, it's important to note that containerd is used by Docker. The Docker engine previously implemented both high-level and low-level runtime features. These are now broken out into separate containerd and runc projects.

Three of the container runtimes we reported last year, rkt, lxc, and mesos, have dropped to nearly undetectable levels. At the same time, CRI-O has made its debut. One thing that surprised us is the small adoption rate to date. CRI-O, a lightweight runtime for Kubernetes, started at Red Hat in 2016 and was adopted into the CNCF® in 2019. We expect its use to climb over the coming years, especially as customers running Red Hat OpenShift migrate from v3 to v4, where CRI-O replaces the previously provided Docker engine.

Which container runtime to choose may seem a little unclear given the emergence of several options. Different solutions cite aspects like reduced overhead, stability, extensibility, and container registry compatibility as advantages. Now, however, because of the open standards, concerns about making the wrong choice and lock-in have evaporated. To make it even easier, popular platforms like OpenShift, GKE, and IKS support using multiple container runtimes in parallel and have typically designed in a runtime of choice, removing the need to spend any cycles on deciding which one to use.



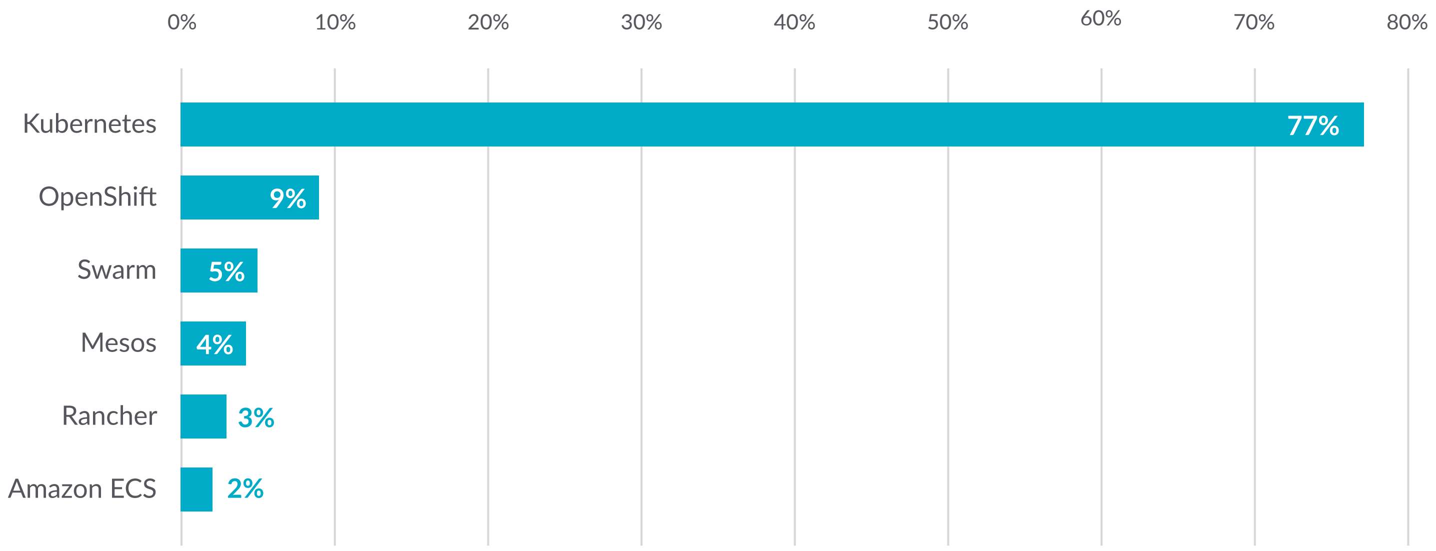
Container orchestration platforms

The headlines about Kubernetes winning the orchestration war have all but disappeared in 2019. It's no surprise that as the de facto container orchestration tool, it takes a whopping 89% share across our customer base when you add in Red Hat OpenShift and Rancher — both built with Kubernetes. The chart at the bottom of the page shows the current breakdown.

Year-over-year, Swarm takes the biggest share drop from 11% in 2018 to 5% in 2019. As noted in last year's report, given Docker's late embrace of Kubernetes in late 2017, we expected the changeover to be forthcoming. Users simply needed enough runway to make the shift.

Mesos use, which includes users with Marathon or DC/OS from D2IQ (the company formerly known as Mesosphere), maintains a steady 4% share. However, like Swarm, given the strategy of D2IQ around Kubernetes, this number is likely to shrink in the next year.

Orchestrators

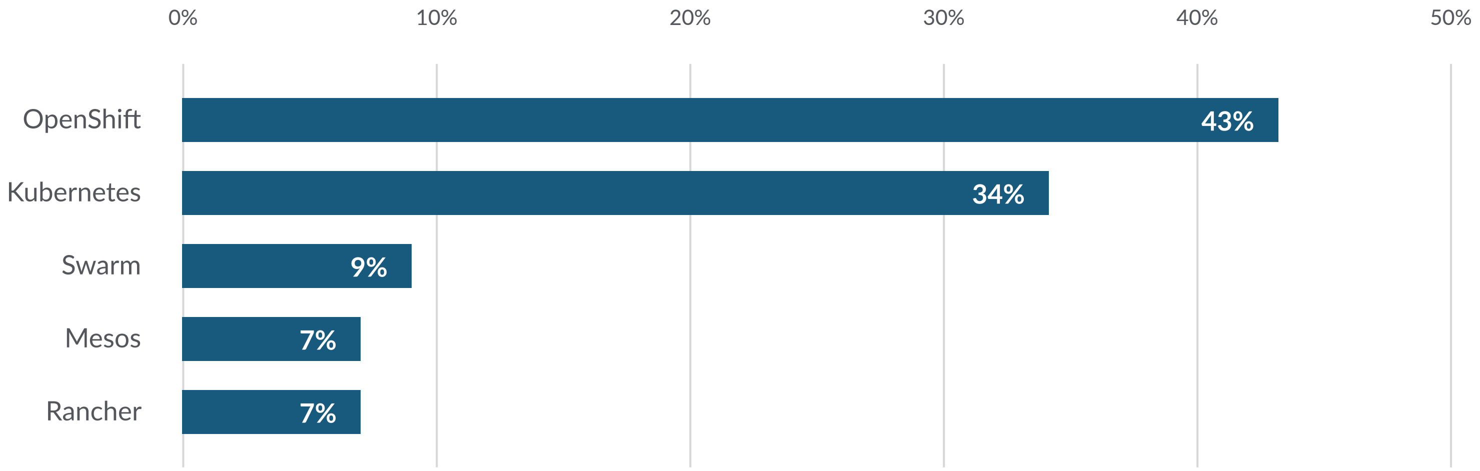


Which platform do on-prem customers choose?

From working with our customers we know that there is a difference in the adoption patterns of larger, more risk averse enterprise customers. When we separate the data for companies who deploy the Sysdig platform on-premises, the picture changes significantly. The Red Hat OpenShift Container Platform comes out on top with this segment. This is primarily because these organizations want the advantages of Kubernetes, but prefer to do so with a commercially supported on-prem Platform-as-a-Service (PaaS) solution like OpenShift.

If you're yet to make the jump into cloud native, you can use this insight to confirm that the right orchestration choice is Kubernetes. All that's left to be done is choosing which "flavor" of Kubernetes is right for you. Before you do, let's look at some data around public clouds.

Orchestrators — Sysdig On-Premises



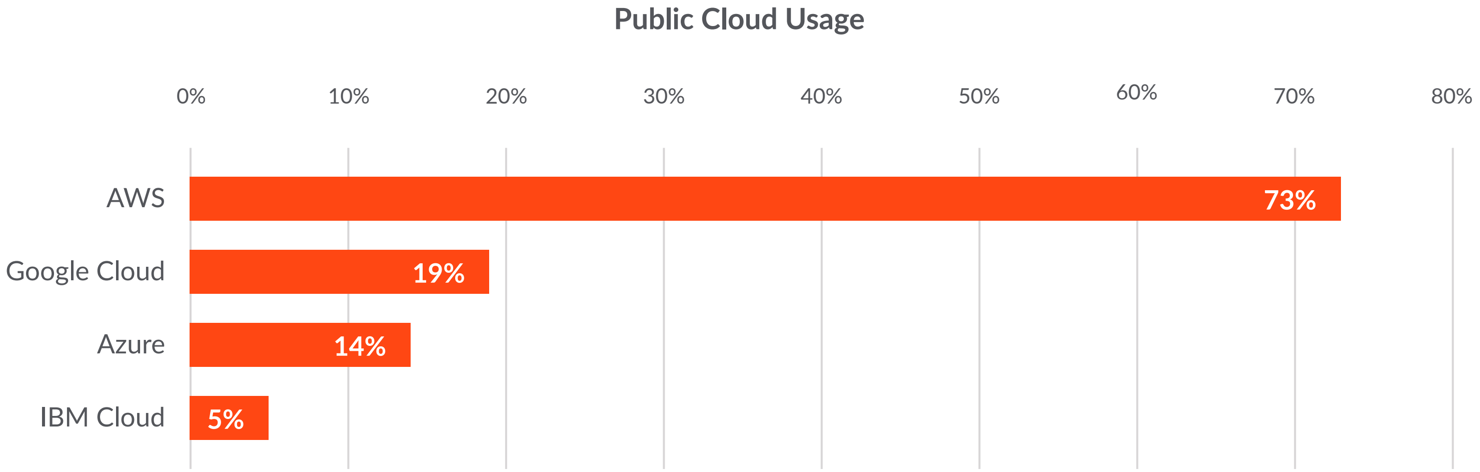
Which public clouds do customers choose?

We segmented which clouds were in use to determine the popularity of the different cloud providers among our customers.

By a large margin, Amazon (AWS) is the public cloud of choice with Sysdig users. Given that AWS holds the largest share of the cloud/IaaS use, it's reasonable to expect that the numbers would align to a similar pattern. Another potential factor in the traction with AWS is that Sysdig has forged a partnership with Amazon that has produced a number of integrations and certifications, including the availability of the Sysdig Secure DevOps Platform on the AWS marketplace. With similar efforts underway with Microsoft and Google, it will be interesting to see what happens with these numbers over the next year.

Reflected in the above as well is the fact that approximately 11% of customers are multi-cloud, meaning they operate and monitor container clusters running in more than one public cloud.

*Note: For this data point we have excluded data from **IBM Cloud Monitoring with Sysdig** since it's currently offered exclusively to public IBM Cloud users.*



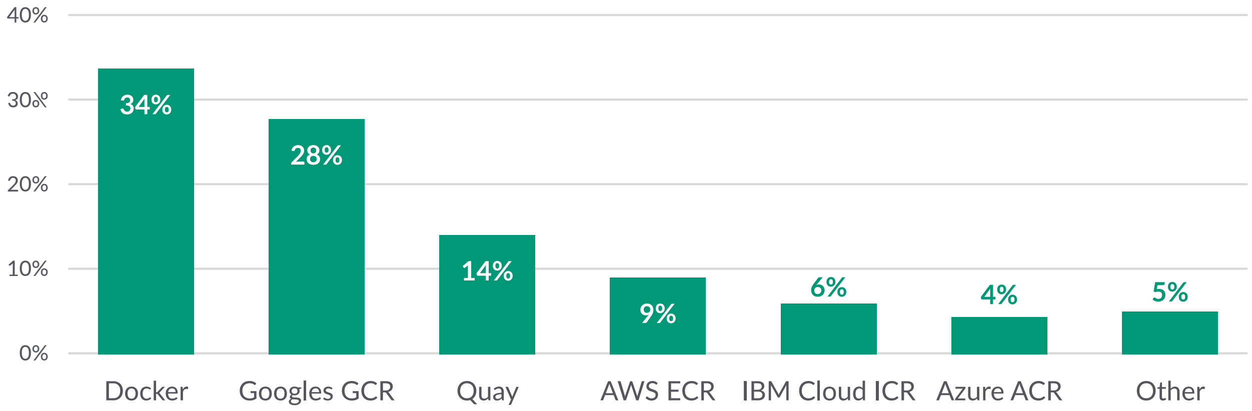
Security and Compliance

As organizations move container workloads to production, they are recognizing the need to integrate security and compliance into the DevOps workflow. “Shift security left” has become a buzz phrase that often refers to scanning containers for vulnerabilities. Scanning is clearly critical given the high percentage of container images pulled from public registries and the high failure rate of scanned images. But the survey data also highlights the need for compliance checks and stringent runtime policies to reduce risk. To provide insights into the state of security and compliance in Kubernetes and cloud-native environments, we’ve analyzed data points that include vulnerability scanning, runtime security, and compliance.

Vulnerability management

Customers scan images to identify, block, and resolve container vulnerabilities within CI/CD pipelines and container registries. Here we look at two data points — the top registries in use, and the success/fail rate when scanning images for vulnerabilities.

Container Registries



Public and hosted container registries

Container registries provide repositories for hosting and managing container images. Docker registries are most frequently used — common within 34% of our customers. This measure includes both private hosted and public repositories. Registry solutions hosted by cloud providers are increasingly popular. Similar to 2018, in 2019, the Google Cloud Registry is again the top public cloud repository, used by 28% of our Sysdig users.

Within these various offerings, we looked at the percentage of containers pulled from public vs. private repositories. We found that 40% of images come from public sources. The risk of using container images from public repositories is that few are validated or checked for security vulnerabilities. Using Docker Hub as an example, images with "Certified," "Official," and "Verified Publisher" are likely trustworthy. However, of the nearly 3 million images hosted, less than 1% carry these designations. To reduce the risk, our customers are creating policies to define which container registries are approved for use in their organizations.

Images Pulled from Public vs. Private Registries

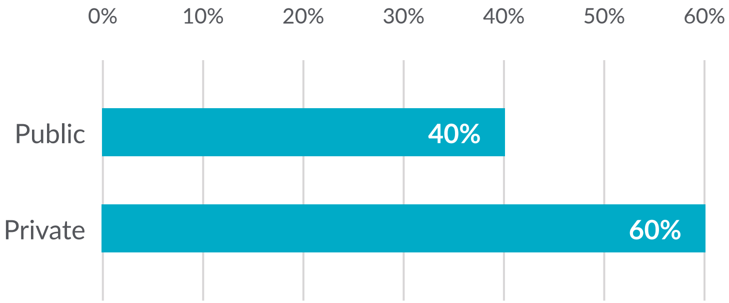


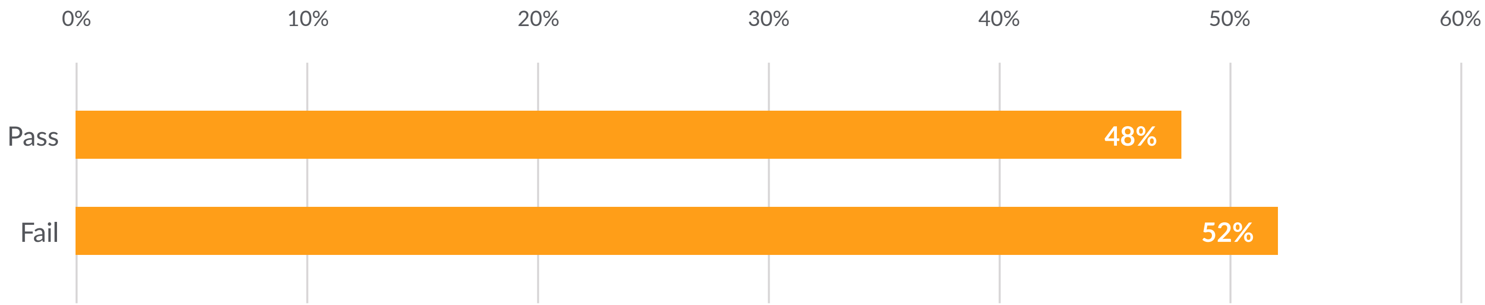
Image scanning

Regardless of the source of the container images, it is critical to perform image scanning and identify known vulnerabilities prior to deploying into production. To quantify the scope of the risk of vulnerabilities, we sampled pass and fail rates for images scanned over a five-day period. Over half of the images failed, meaning they were found to have known vulnerabilities.

> "We need to check configurations and validate that our images are free of vulnerabilities before pushing to production."
>
> –
>
> Global Travel Company

Scanning Results

Median of Containers Scanned



Runtime security threats

Once known vulnerabilities have been addressed in the build phase of the container lifecycle, teams need to set policies that will detect anomalous behavior and trigger security alerts at run time. Runtime security for Kubernetes is something organizations are just starting to address. Falco, the CNCF open-source project contributed by Sysdig, is quickly gaining momentum and interest. In the last 12 months, there have been over 6.7 million Docker Hub pulls, an increase of 252% over the prior year. Falco enables the definition of runtime policies that detect security violations and generate alerts. As users adopt Falco, they are using Sysdig Secure to automate rule creation and tuning.

> **"With security events, the frontline is our developer team. They know what their applications should and should not be doing."**
>
> —
>
> Director of Engineering at a Global Travel Company

Top runtime policy violations

We looked at policy violations as measured by the volume of alerts customers are receiving. This indicates the types of runtime security risks that container users are uncovering most frequently. Each of the following violations are detected by Falco security policies that are enabled by default in Sysdig Secure. Below, we provide the top 10 violations in order of frequency, along with a description of each to explain the possible threat.

Violation	What it is	Why it's a security threat
Write below etc	Attempt to write to any file below the /etc directory	Adding or altering files in /etc, could be an attempt to change the application behavior.
Write below root	Attempt to write to any file directly below / or /root	Modifying data in these directories could be an unauthorized attempt to install software on the container.
Launch privileged container	Starting a privileged container	Privileged containers can interact with host system devices, cause harm to the host OS, and gain access to other containers.
Change thread namespace	Attempt to change a program/thread's namespace by calling setns	Could indicate a privilege escalation and an attempt to gain access to other containers.
Launch sensitive mount container	Starting a container that has a file system mount from a sensitive host directory	Indicates the container has to access to data volumes that might contain sensitive files.
Non sudo setuid	Attempt to change users by calling setuid	Could indicate an attempt by a process to elevate its privileges.
Write below binary dir	Attempt to write to any file below a set of binary directories	Could indicate a malicious attempt to install unauthorized software like backdoors.
Run shell untrusted	Attempt to spawn a shell below a non-shell application	Enables an attacker to manipulate the system, download malware, or initiate other malicious activity.
System procs network activity	Network activity performed by system binaries that are not expected to send or receive network traffic	Binaries that are should not have network activity have network activity, indicating that the binary has been compromised.
Terminal shell in container	A shell was used as the entrypoint/exec point into a container with an attached terminal	Enables an attacker to manipulate the system, download malware, or initiate other malicious activity

Compliance

Since today's enterprises face a number of governance and regulatory compliance requirements including PCI-DSS, HIPAA, and GDPR, taking steps to follow best practices in order to comply with regulations is imperative.

The Sysdig platform runs compliance checks against monitored clusters to check hosts, containers, and other aspects of the environment against a defined set of best practices. This includes the Center for Internet Security (CIS) benchmark tests, CIS benchmark for Kubernetes and CIS benchmark for Docker.

We chose a sample from over 80 benchmark rules from the CIS benchmark for Docker to highlight the state of compliance against these best practices with Sysdig users. The seven benchmarks evaluate container images residing on each host for configuration issues related to permissions, security tooling, and capabilities that have the potential to expose an organization to risk.

We took the median score for each of these six container checks. The score, in this case, is the measure of containers per host that fail the test and do not adhere to the recommended best practice for reducing risk.

> "Troubleshooting, forensics and audit can be handled at scale when you have a single source of truth across the teams."
>
> –
>
> VP of engineering at a top 5 investment bank

Benchmark	Median number of vulnerable containers per host	Why it's a threat
Containers with default seccomp profile disabled	31	The secure computing mode (seccomp) of the Linux kernel provides a system call filter that restricts the actions available within a container. If disabled, there is an increased risk of attacks via unrestricted system calls.
Containers with no AppArmor profile	28	An alternative to SELinux, AppArmor is available by default on most Linux distributions. AppArmor enables the association of a security profile to each application and restricts access to the underlying system.
Containers without restricted privileges	28	By default, Docker starts containers with a restricted set of capabilities. If unrestricted, capabilities can be used to escalate privileges or for container breakout.
Containers running as root	21	Base container images typically ship with the default user set to run as root to allow for custom package installation. A non-root user should be added to the Dockerfile during image build or specified at runtime. Failing to do so exposes the potential for privilege-escalation attacks from within a container.
Containers with no SELinux security options set	6	SELinux — enabled by default with Docker — labels every process, file, resources, etc., for security context to enable rule-based control of access rights. If disabled, hosts and other containers are exposed to undesired activity and access.
Containers running in privileged mode	4	Privileged containers run with all capabilities enabled, exposing the system to the risk of privilege escalation and container breakout.

We aren't able to determine from our data the reasons behind why a large number of containers are not using some of the default security tooling like seccomp or AppArmor. The large number of containers running as root is possibly due to the fact that default users for images is set to root, making it easy for a service to inadvertently run with unrestricted privileges.

What Services are Customers Running?

The top 10 open-source solutions running in containers

Open source 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 10 open source technologies deployed by Sysdig customers:

NGINX	redis	elastic	node JS	PostgreSQL
60%	21%	14%	14%	14%
GO	APACHE	Java	RabbitMQ	mongoDB
14%	12%	12%	11%	11%

The 2019 list includes a wide range of services — each critical to the function of modern applications, including:

- HTTP server and reverse proxy solutions — NGINX and Apache
- NoSQL, relational, and in-memory database solutions — MongoDB, Postgres, and Redis
- Logging and data analytics — Elasticsearch
- Programming languages and frameworks — node.js, Go, and Java/JVMs
- Message broker software — RabbitMQ

Given the wide range of options available in the open source community, it's surprising that the services in our list have remained fairly consistent over the past three years. This year, we purposely omitted Kubernetes components like etcd and fluentd. Since these are deployed by default, they end up at the top of the list for every Kubernetes user.

What's new this year is the arrival of both Node.js and Go (aka golang) overtaking the use of Java. Java has long been one of the most prominent programming languages, but newer options like Go, created by Google engineers, have gained favor with DevOps and Cloud teams in part because of their ease of use. Node.js, a JavaScript runtime, simplifies writing code that runs equally well on servers as well as browsers. It is well suited for the new generation of databases like CouchDB and MongoDB, which support queries written in JavaScript.

The top 10 solutions above are widely deployed and trusted services. If you're in the market for similar services, you can't go wrong with taking advantage of what these open source solutions offer. There is, however, a long tail of software solutions available.

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 clouds. Of the three mainstay solutions, JMX, StatsD, and Prometheus, the past year saw Prometheus rise as the top solution in use.

Year-over-year, Prometheus metric use increased 130% across our customers — up from 20%. As the use of new programming frameworks expands, alternatives like JMX metrics (for Java apps) and StatsD are diminishing, down 45% and 17% respectively.

One of the most successful open-source projects to emerge from the CNCF, Prometheus has become synonymous with cloud-native monitoring. It is now widely adopted as a metric standard in projects like Kubernetes, OpenShift, and Istio. In addition, an increasing number of "exporters" are available to provide metric output for a wide range of third-party solutions.

We expect the popularity of Prometheus to continue its growth within our customer base, particularly as Sysdig extends its offering of Prometheus compatible monitoring focused on large-scale environments. In addition, with the start of the OpenMetrics project based on the Prometheus exposition format, it will likely be an additional catalyst for the Prometheus approach to metrics.

	Prometheus	JMX	StatsD
2019	↑ 46%	↓ 30%	↓ 24%
2018	20%	55%	29%

Top Prometheus metrics and exporters

Diving deeper into Prometheus, we wanted to understand the types of data that cloud teams export into Sysdig most often.

The metrics in the following chart showcase the most frequently used metrics and exporters.

Top 10 Prometheus metrics and exporters

Metrics and exporters	% of Prometheus and Sysdig users	What it monitors
process	93%	Process metrics from /proc including CPU, bytes written or read, number of processes, and page_faults
http	67%	Http metrics like request count, duration, response statuses
nodejs	54%	Custom metrics instrumented with the Prometheus nodejs library
go	54%	Custom metrics instrumented with the Prometheus go library
python	13%	Custom metrics instrumented with the Prometheus python library
grpc	12%	Metrics from gRPC, an open-source remote procedure call system initially developed at Google
etcd	10%	Metrics from etcd including the status of the etcd server, disk operations, network, and processes
JVM	10%	Java virtual machine metrics including heap, thread, and garbage collection
jaeger	10%	Metrics for the Jaeger distributed tracing system including trace and span counts, latency, and errors
istio	8%	Metrics generated by the Istio service mesh including request and TCP metrics with relevant labels

It’s no surprise that process is at the top of the list as is the default for Prometheus libraries. The large percentage of use for nodejs and go confirm the rise of these programming languages, and indicate that Prometheus metrics are the favored way to monitor their performance. We expect Istio, currently at the bottom of the list, to grow in usage over the next 12 months as service mesh solutions gain greater adoption.

Those who know Prometheus well may wonder why three exporters in particular are not on the list: Node exporter, kube-state, and cAdvisor. The metrics these exporters generate for hosts, Kubernetes, and containers, respectively, are all available natively from Sysdig.

> "For a lot of the application containers we run on Kubernetes, Prometheus metrics are what look at first to know if things are running as expected."
>
> –
>
> DevOps engineer at a healthcare company

Containers

Each year, we take a look at details specific to the count and activity around containers, including density and lifespans. This provides insight into the rate of adoption but also illustrates the scale and efficiencies being achieved.

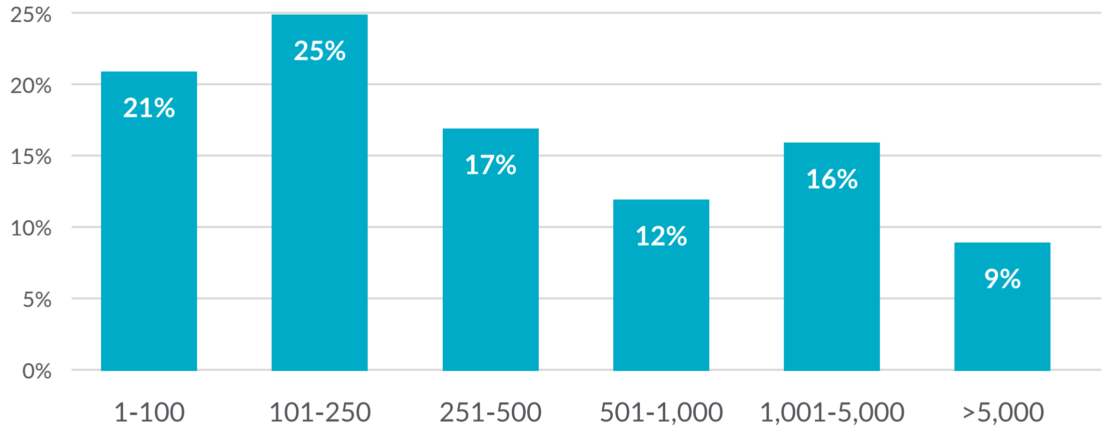
Containers-per-organization

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.

Nearly half of customers run 250 containers or fewer containers. At the high end, 9% of customers are managing more than 5,000 containers.

While working with containers and Kubernetes is old hat for many in the open-source world, some enterprise customers are only beginning to take their first steps into the new world. It is common for adoption to begin at a small scale, sometimes born from developers who push for containerization as a means to accelerate software delivery. But tiger teams, increasingly initiated by innovation, are tasked with leading their organization into the cloud-native era. DevOps and cloud teams report that once the benefits are proven, adoption accelerates as more business units look to onboard to the new platform.

Number of Running Containers



Container density

Containers-per-host density increases 100%

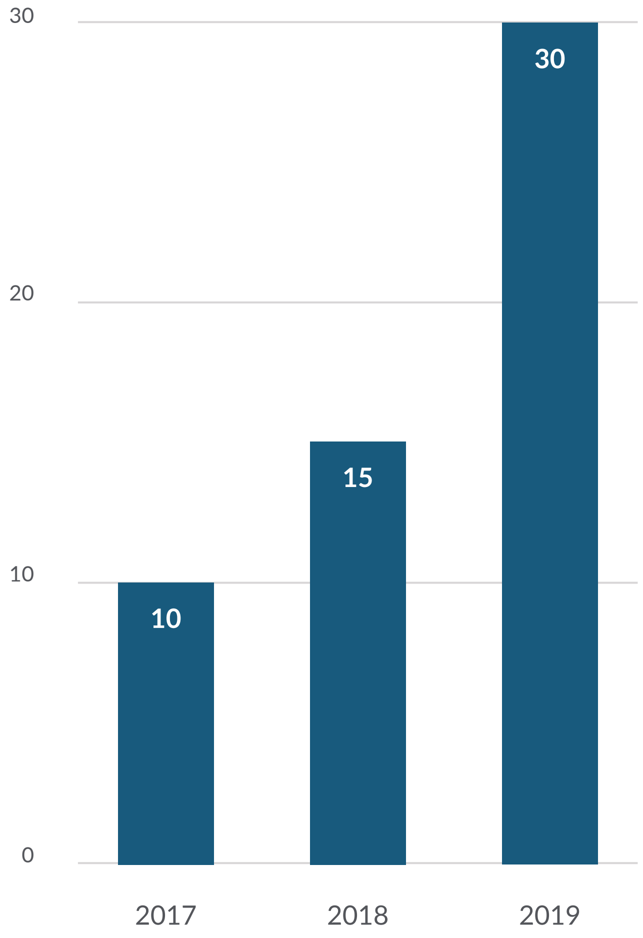
Over the past year, the median number of containers per host doubled to 30, compared to 15 in 2018. We expected this number to increase based on several factors:

1. Growth in the number of applications being transitioned to cloud-native infrastructure
2. Inclusion of data from on-premises Sysdig customers who run larger, denser clusters
3. Increases in compute "horsepower," enabling more containers to run on each node

This metric is likely to continue upward. For 2019, the maximum per-node density we saw was 250 containers — a 38% increase from 2018.

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. In addition, our customers report that with the transition to containers orchestrated across a cluster of nodes, they are able to extend the life of existing hardware. As a result, our customers have less concern about the impact of aging hardware on application uptime and performance.

Median Containers per Host



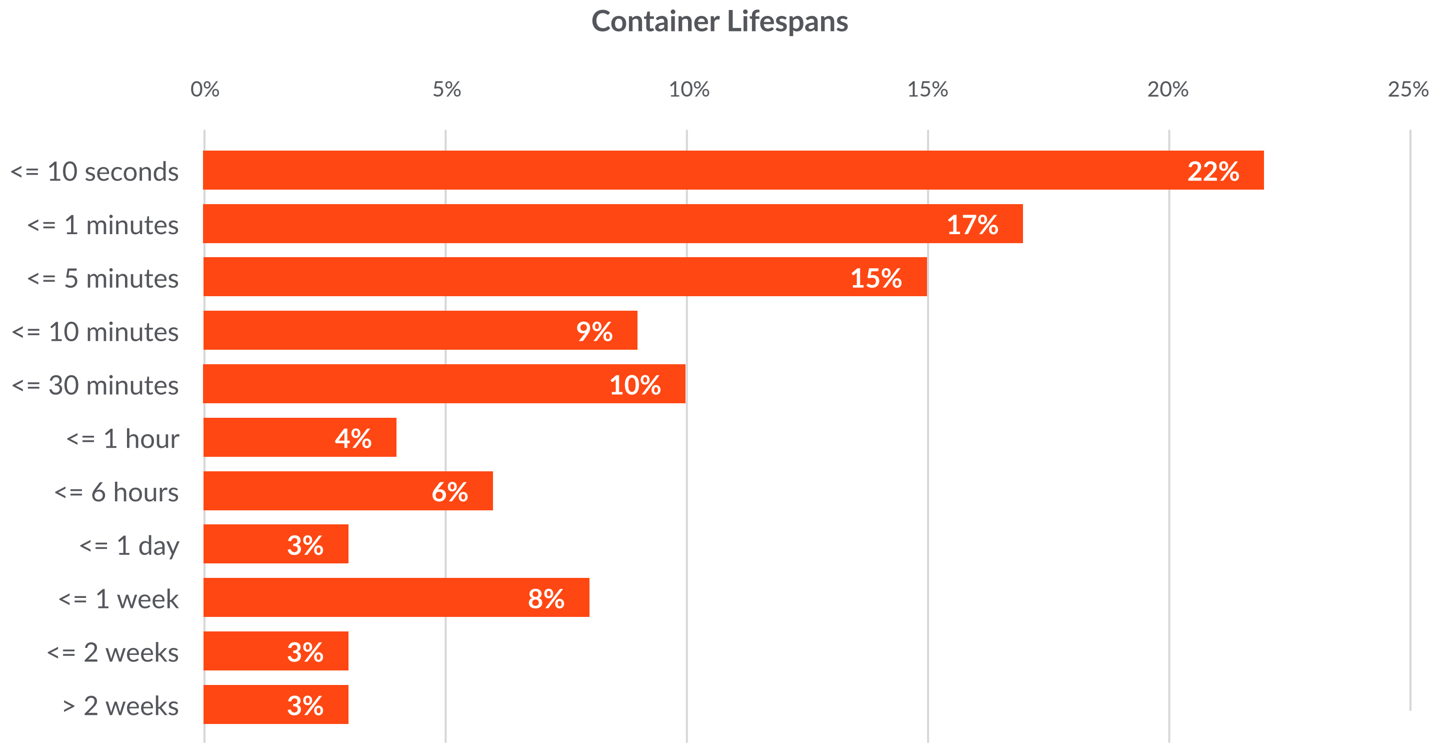
Container, image, and service lifespans

The measure of how long (or how short) containers, container images, and services live was one of the most popular data points from our 2018 report. It reflects just how dynamic modern applications are from both a development and a runtime perspective.

The short life of containers

Comparing container lifespans year over year, we see a similar pattern where a majority of containers are alive for less than a week. In fact, our newest data sample shows that the number of containers that are alive for 10 seconds or less has doubled to 22%.

At one week, there is a spike in containers stopping — 8%. We investigated why this might be the case and found that we can correlate this to Kubernetes doing its job of auto-scaling up and down. During the weekend, as demand on services decrease, Kubernetes reduces number of running instances per service.



Many containers need to only 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. We expect the number of containers with short lifespans to increase, especially on serverless platforms that are well-suited to running short term tasks.

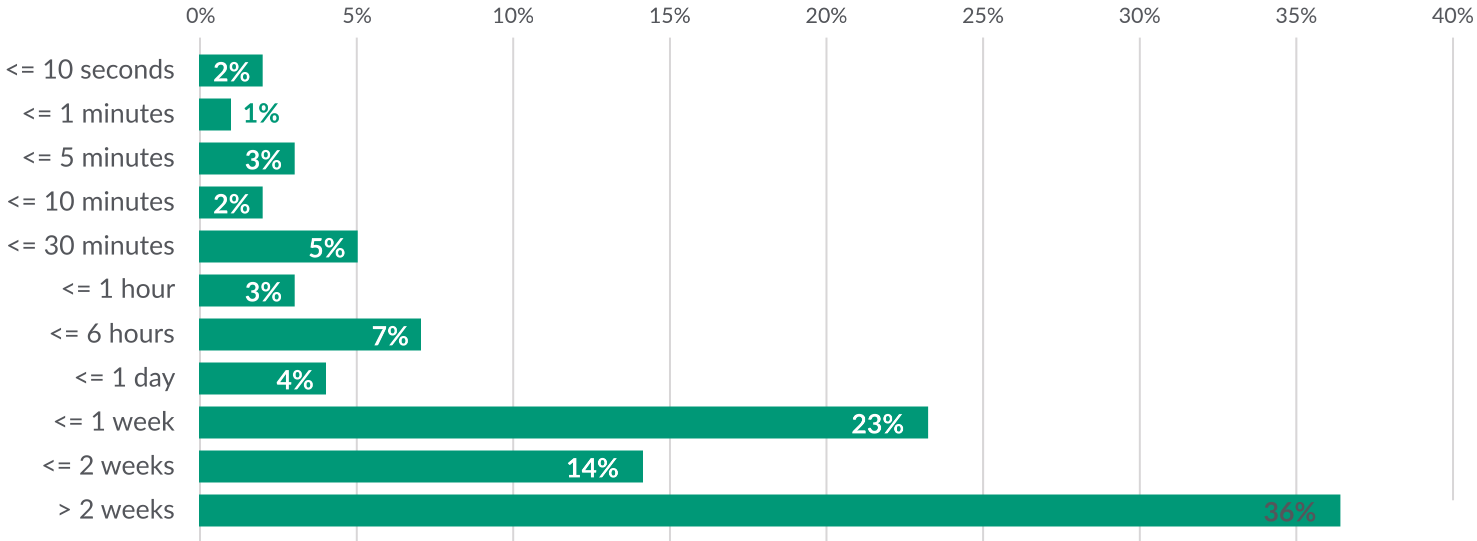
The ephemeral nature of containers is one of the technology's unique advantages, yet at the same time can be a challenge in seeing issues around security, health, and performance. For this reason, along with adopting containers and orchestration, other pieces of the value chain like monitoring, security, and compliance tooling also need to be reevaluated.

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 over half of container images get replaced — also known as churn — in a week or less. For most if not all of today's businesses, speed to market matters and makes all the difference in maintaining competitiveness. Code deployment is being deployed more frequently, which in turn means new container images. Containers support what businesses need to turn great ideas into reality, fast.

Container Image Lifespans

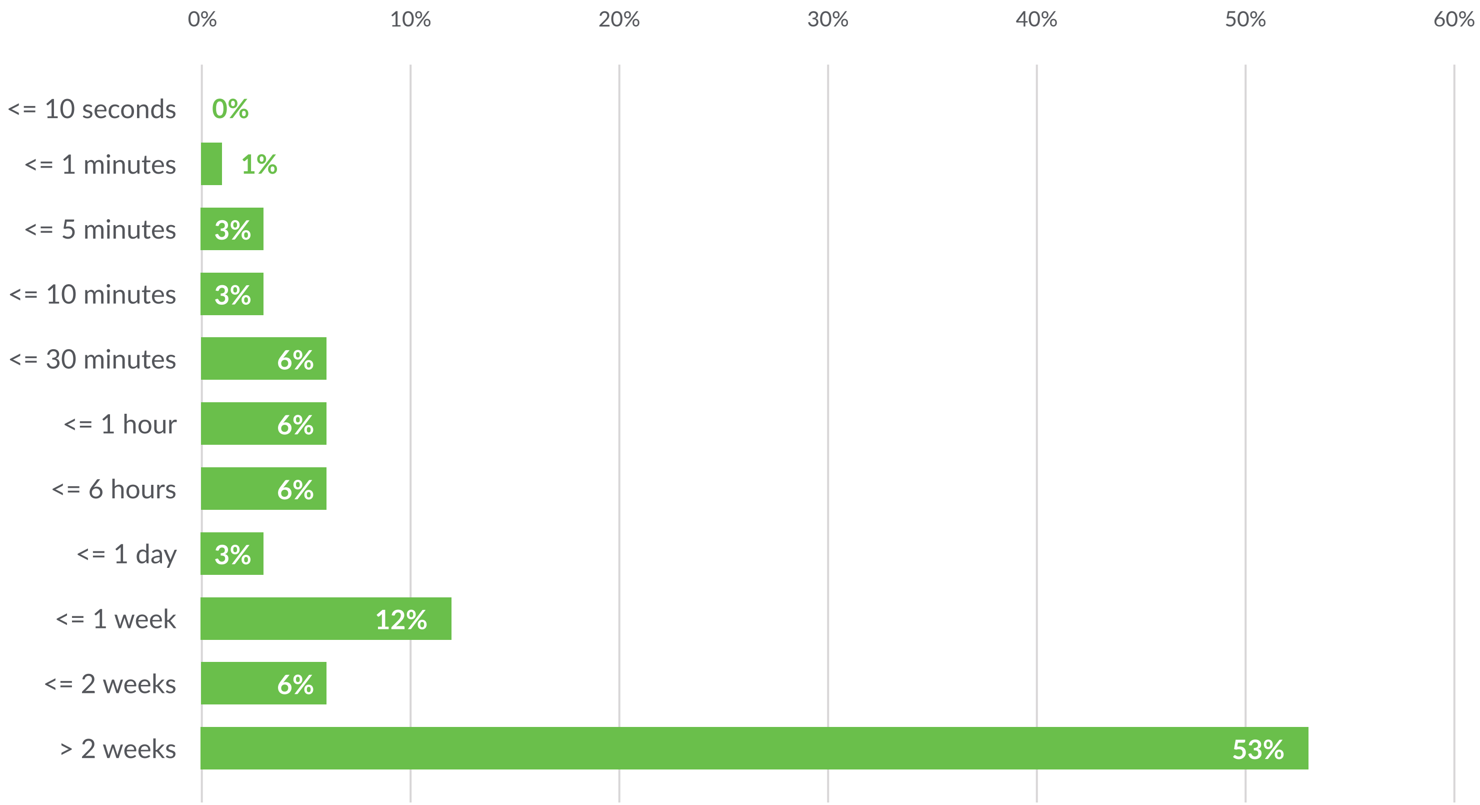


Service uptime

For our last view into lifespans, we examined the data around services and uptime. Services — the functional software components of our applications like database software, load balancers, and custom code — might be continuously improved, However, at the same time, it's important (at least for most 24/7 businesses) to keep services up and running around the clock.

Similar to 2018, over half of our customer services are up and running non-stop for more than two weeks. Underneath, containers will start and stop to support scaling and other operations, but applications will remain up. With the increase in the frequency of code releases, containers and solutions — like Istio — help smoothly execute rolling or canary deployments without impacting your services.

Service Lifespans

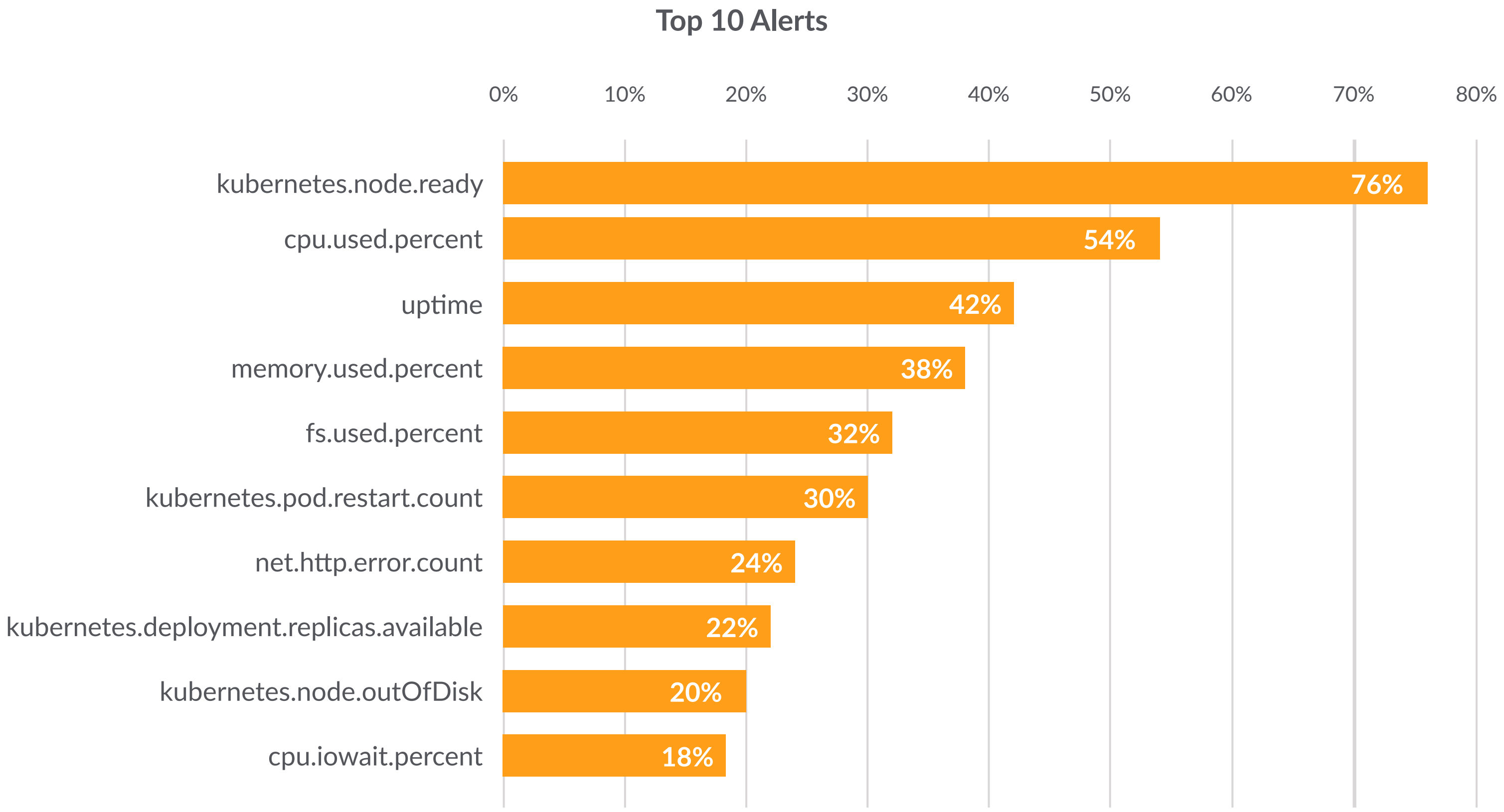


Alerts

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. The makeup of these alerts has changed since our last report, shifting in favor of Kubernetes infrastructure while continuing to focus on resource utilization and uptime.



Alert scopes

Sysdig alerting supports customization by "scoping" to a specific tag or Kubernetes / cloud label. For instance, using an example from the above alerts, you can specify memory.used.percent alert for an individual namespace like "istio-system", or for a specific Pod name like "envoy" inside that namespace. Tagging and labeling play a critical role in cloud-native environments, providing unique identifiers that help organize and isolate items. In this case, the tagging specifies a group of "things to watch."

Specifying alerts by Kubernetes labels is now one of the most common practices, including namespace, cluster, Deployment, and Pod in the top five. Agent tags — the metadata attached to the Sysdig agent when deployed — rise to the second most popular alert scoping across Sysdig users.

The results in 2019 indicate that alerts are now set by application rather than by host, followed by microservice (Deployment).

2018

Scope label type	% of users
Kubernetes Pod name	70%
Kubernetes namespace	68%
Host name	62%
Container name, image, or ID	39%
Cloud provider tags	28%

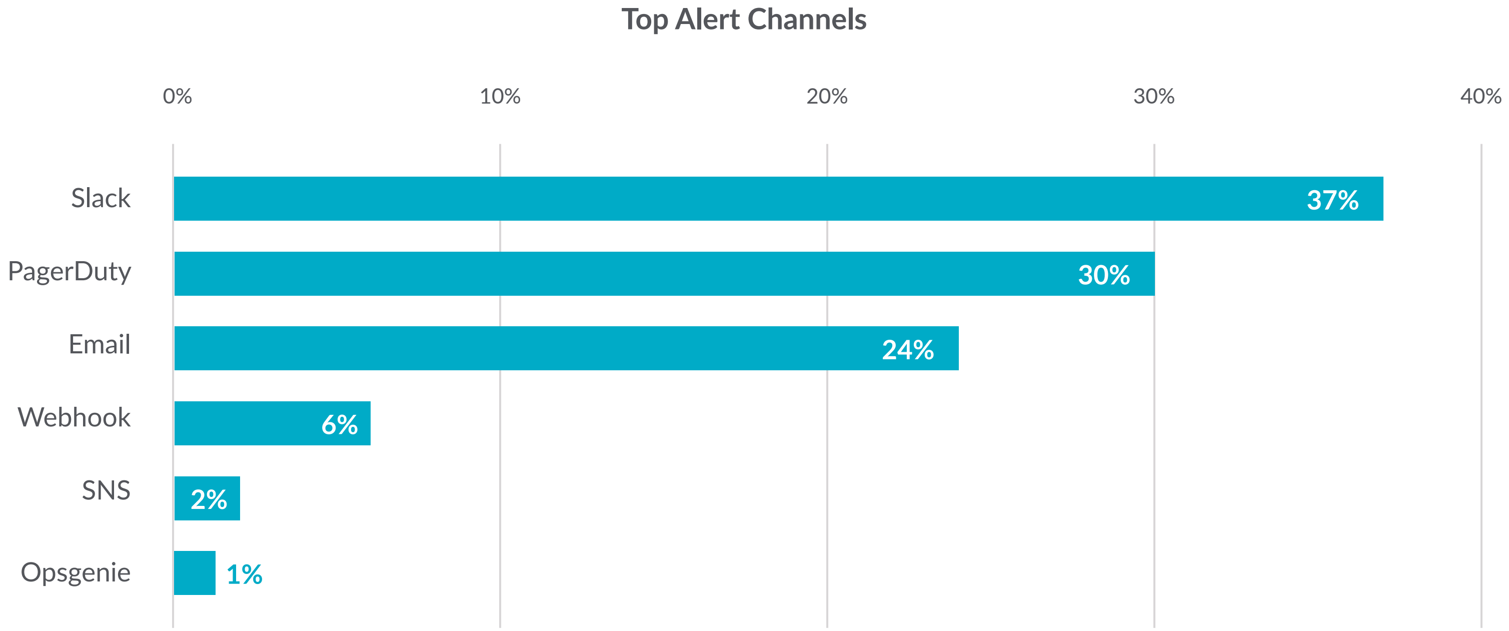
2019

Scope label type	% of users
Kubernetes namespace	94%
Agent tags	78%
Kubernetes cluster	76%
Kubernetes Deployment	71%
Kubernetes Pod	38%

Alert channels

We looked at the communication channels users have configured to receive alerts. Slack took the top position, greater than purpose-built incident response platforms and even email.

We find the results interesting because unlike PagerDuty and Opsgenie, for instance, Slack is not considered an incident response platform. It's likely that Slack is being used for non-critical alerts handled during working hours while solutions like PagerDuty are being used for "waking people from bed."



Kubernetes Usage Patterns

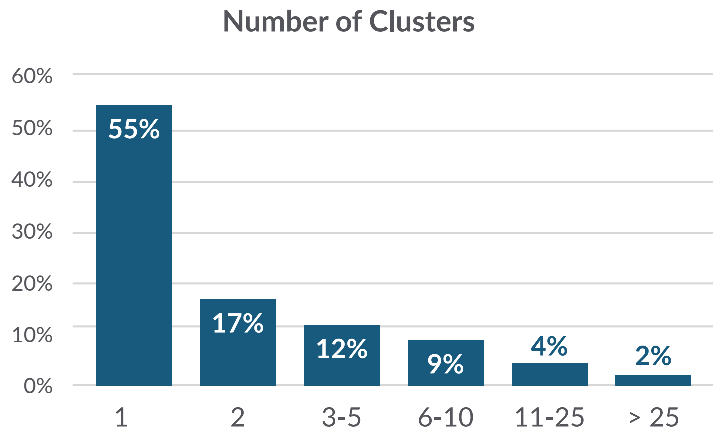
How many clusters are customers operating? How many Pods run per node? Does anyone use Kubernetes Jobs? In this section, we answer these questions and more. We look at a range of details about what customers are doing with Kubernetes from clusters to ReplicaSets.

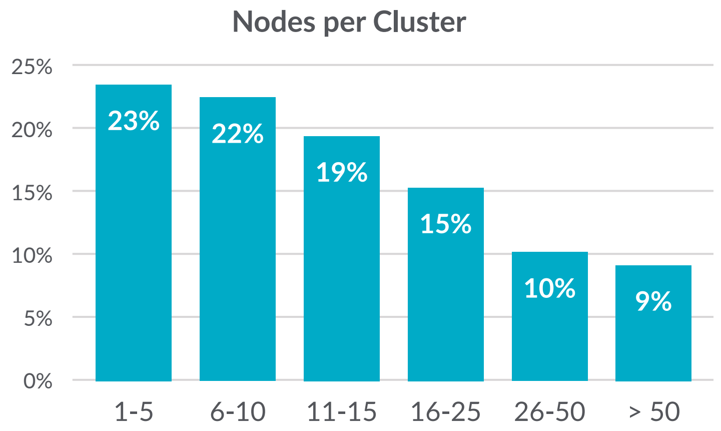
Because Sysdig automatically collects Kubernetes labels and metadata, we're able to provide cloud-native context for all of the data insights we discover from performance metrics and alerts to security events. This same capability enables us to capture each of the following usage metrics from the cluster all the way to Pods and containers, all with a simple query.

Kubernetes clusters and nodes

Some customers maintain a few clusters — some small, some large — while others have a sizeable estate of many clusters of varying sizes. The charts to the right provide 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. WIth the services like Amazon Elastic Kubernetes Service (EKS), Google Kubernetes Engine (GKE), Azure Kubernetes Service (AKS), and IBM Cloud Kubernetes Service (IKS) users can spin up and tear down clusters quickly as needed.





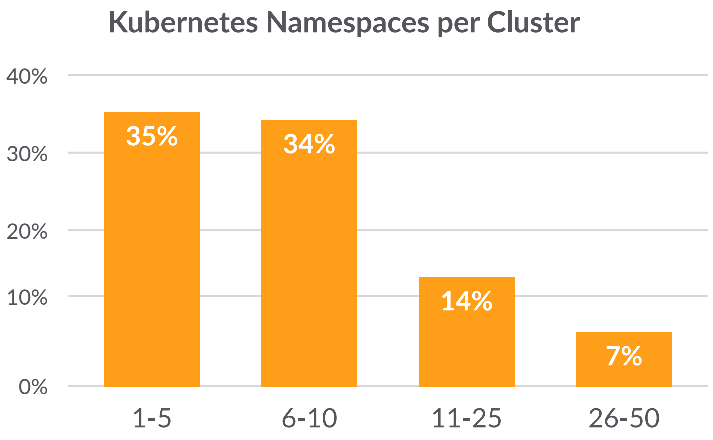
Kubernetes namespaces, Deployments, and Pods

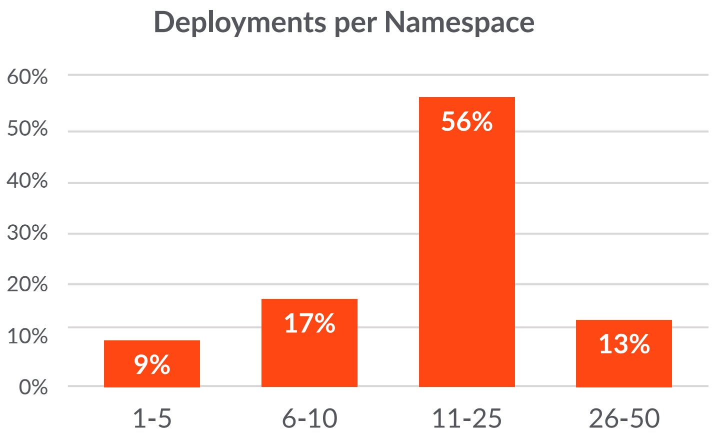
Namespaces per cluster

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 kube-public. 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.



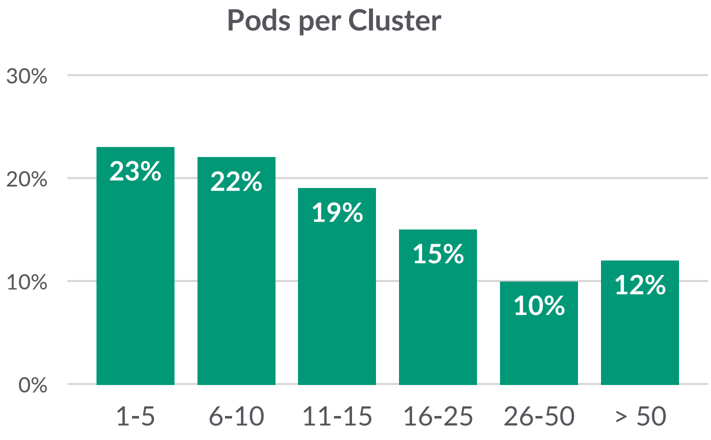


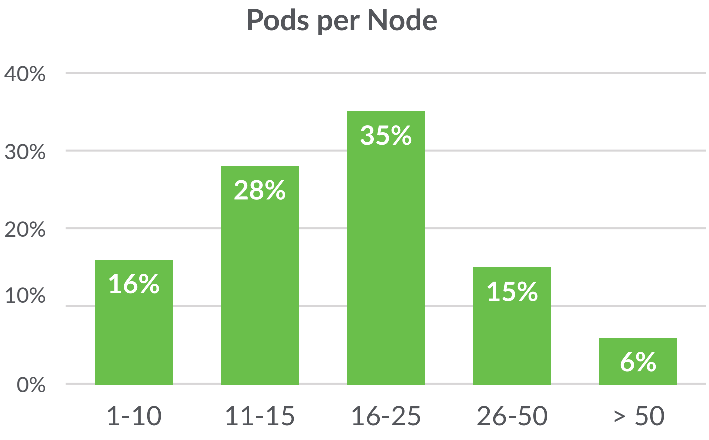
Pods per cluster

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.





StatefulSets and Jobs

In addition to the above look at the Kubernetes objects used with most clusters, we wanted to determine usage of two additional Kubernetes abstractions — StatefulSets and Jobs.

Kubernetes StatefulSets manage the deployment and scaling of Pods that run stateful applications and save data to persistent storage. Unlike standard Pods, StatefulSets provide guarantees about ordering and ensure unique, persistent identities and stable hostnames. Common uses for StatefulSets are to run databases like MySQL or MongoDB.

When containers first started to gain popularity, they were used primarily for stateless applications due to their ephemeral nature and challenges with persistent storage. As the market has matured, many of these challenges have been addressed and the amount of stateful applications running in containers is increasing. Today, 57% of the clusters monitored by Sysdig run StatefulSets.

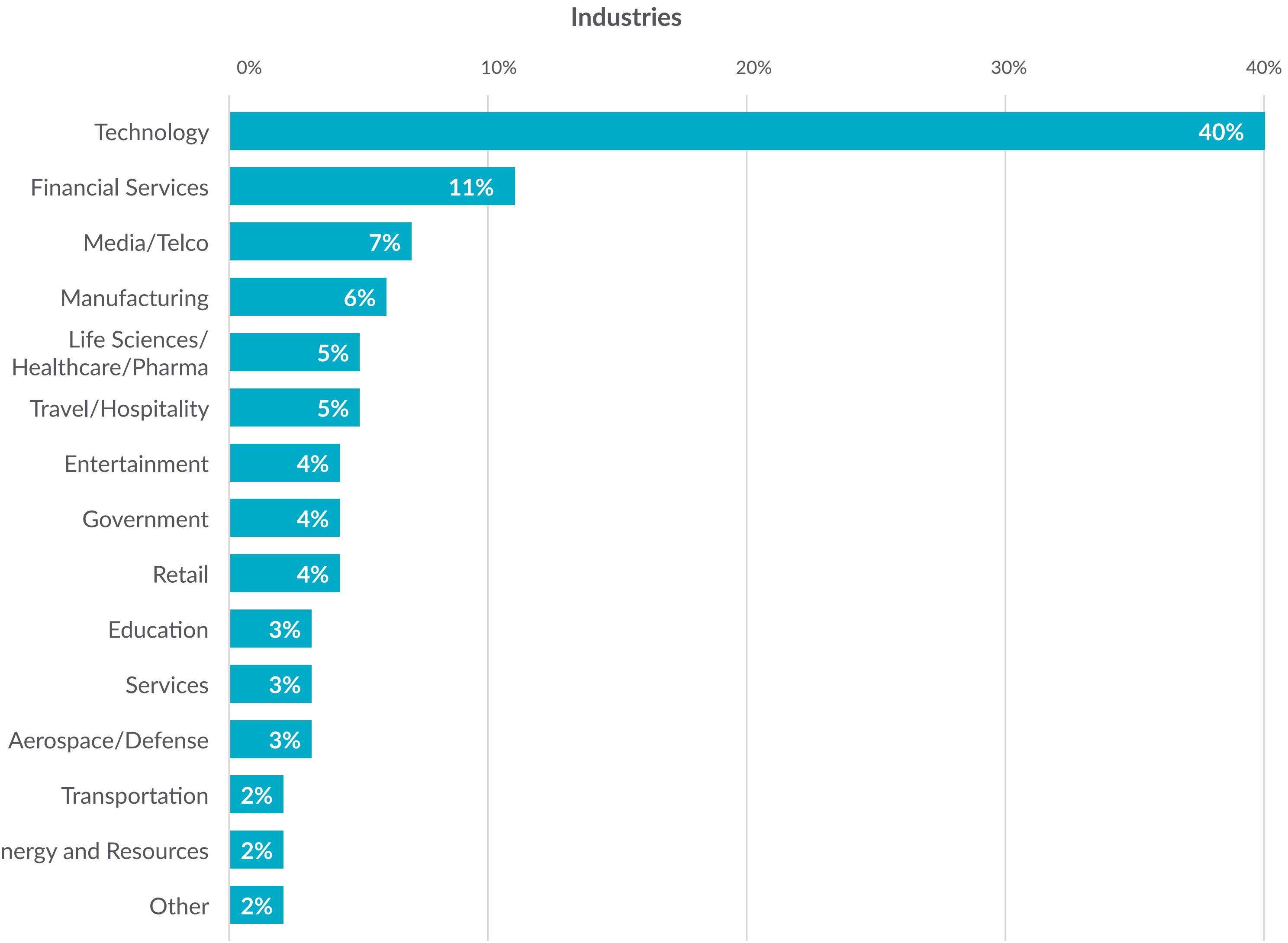
% of clusters running StatefulSets	57%

Kubernetes Jobs create one or more Pods to run a finite task like a batch process and ensure the specific job is completed. Tasks like log rotation, database backups, and running test suites are examples of the types of work that is well suited to run as Jobs. Jobs are also effective for running parallel or sequential processing related work items and are increasingly used to run temporary tasks on serverless platforms. Jobs are currently used on 46% of the clusters we monitor. We've seen one of our customers run over 3000 unique Jobs in a 24-hour period.

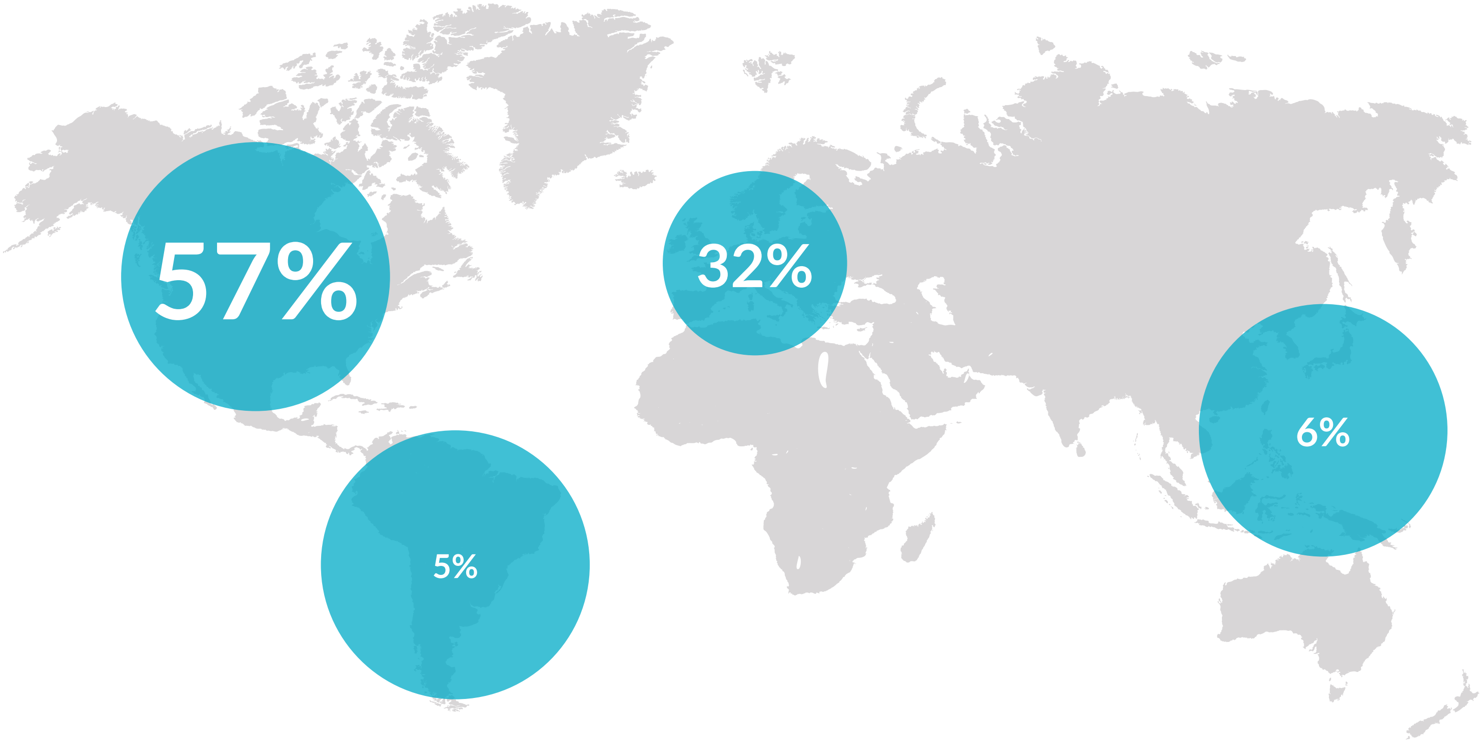
% of clusters running Jobs	46%

Firmographics

The data in this report originates from container deployments across a wide range of industries, with organizations ranging in size from mid-market to large enterprise.



Regions



Conclusion

Container technologies continue to expand their role in transforming how organizations deliver applications. With container density doubling since our last report, it's evident that the rate of adoption is accelerating and as usage matures. The key insights from our third annual report highlight the need for enterprises to take steps to prepare for the massive growth expected:

- Kubernetes is the clear orchestrator of choice, helping organizations deliver applications faster than ever. To keep pace, organizations should invest in Kubernetes-native tools to simplify operating at scale.
- Container environments are more dynamic than ever, with lifespans of 10 seconds or less becoming increasingly common, emphasizing the need for real-time visibility that delivers detailed audit and forensics records.
- Runtime security policies are detecting serious security risks. To keep ahead of these challenges, cloud teams must act now to integrate security into DevOps and shift security left to address risks.
- As Prometheus extends its lead as the standard for cloud-native application metrics, users must learn how to take advantage of it's value reliably and at scale.

Thank you for reading the Sysdig 2019 Container Usage Report. We look forward to following and documenting the evolution of the container market in the coming year. See you then!

Learn how you can confidently run cloud-native workloads in production using the Sysdig Secure DevOps Platform.

www.sysdig.com/platform