# 5 Best Practices to Securing Cloud and Containers

As container and cloud adoption accelerates, most enterprises struggle with visibility into their container and cloud environment. According to Gartner, "Nearly all successful attacks on cloud services are the result of customer misconfiguration and mistakes." They also predict that through 2023, at least 99% of cloud security failures will be the customer's fault.[1]

In addition, containers are essentially black boxes. It's hard to see what's going on inside, and the lifespan of a container is very short. In fact, 49% of containers now live less than five minutes[2], according to our research. Traditional security tools can't see inside containers, handle the dynamic nature of Kubernetes, or scale across multi-cloud deployments. Proprietary security tools can't keep up with the standardization and speed of innovation possible with open source software.

How can you automate security and compliance controls to implement an efficient and secure DevOps workflow? With the right set of integrated tools, you can efficiently manage cloud and container security risks.

It is important to reduce your risk from cloud misconfigurations, continuously scan for cloud and container vulnerabilities, detect abnormal activity, and prioritize threats to ensure your cloud resources and applications are secure across their entire life cycle. These five key workflows will enable you to cover the most critical security and visibility requirements so you can confidently run containers, Kubernetes, and cloud.

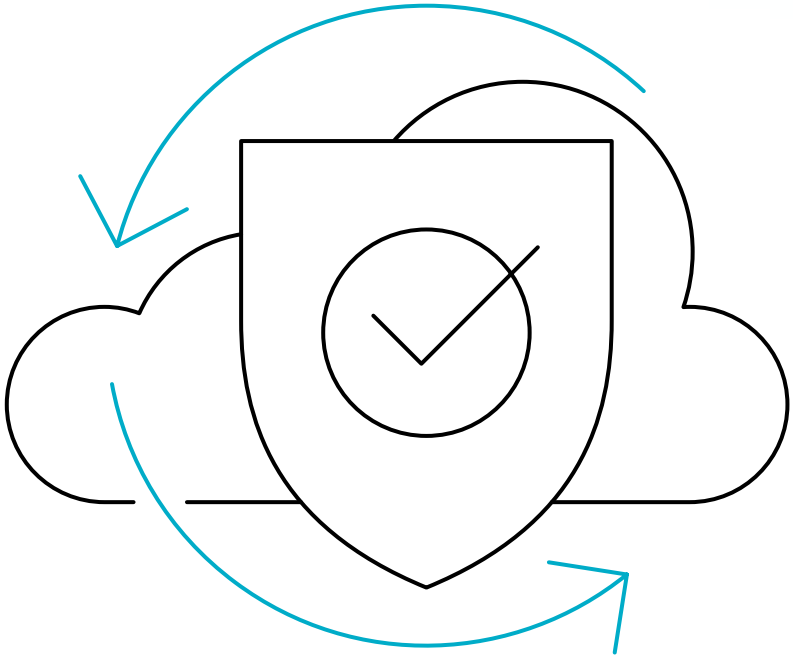
[1] Gartner: Innovation Insight for Cloud Security Posture Management

[2] 2023 Sysdig Container Security and Usage Report
https://dig.sysdig.com/c/pf-2023-cloud-native-security-and-usage-report?x=zp_mX7

1

# Continuous Cloud Security

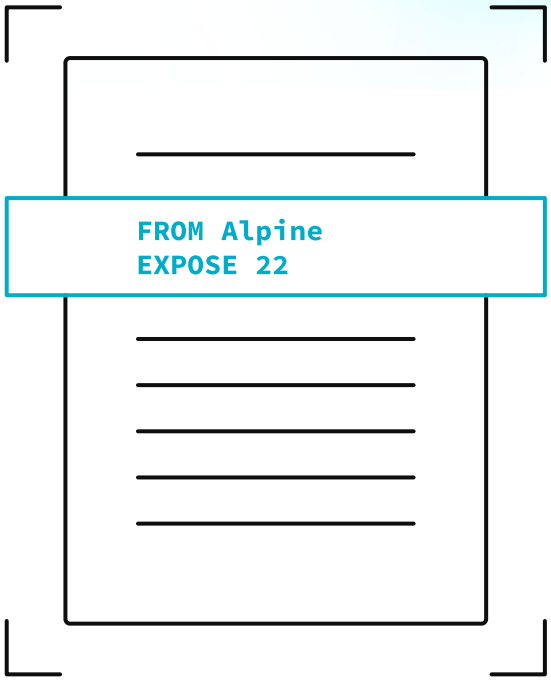
Continuous cloud security is required to immediately identify configuration errors and suspicious behavior. The following steps can help you validate your cloud security posture.



- Improve visibility with an inventory of your cloud resources across multi-cloud environments.
- Improve your security posture by checking your cloud configuration periodically against CIS benchmarks (e.g., public storage buckets, exposed security groups and access controls, etc.) and take steps to remediate violations.
- Increase automation to save time on misconfiguration fixes, integrating remediation into your GitOps workflow.
- Standardize security controls across environments and apply policies consistently with a shared policy model preferably.
- Increase efficiency by prioritizing the fixes that remediate the most security issues around misconfigurations, excessive permissions, or weak controls.
- Reduce drift mapping misconfigurations in production to infrastructure as code (IaC) manifests.
- Detect unexpected changes and suspicious activity across all cloud accounts, users, and services by parsing cloud activity logs.

2

# Prioritize vulnerabilities based on runtime intelligence

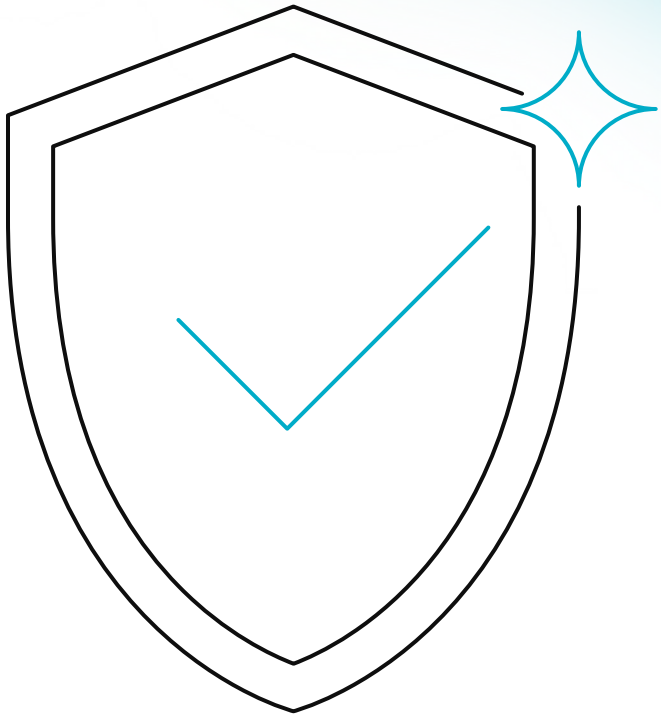



As application development speeds up with CI/CD pipelines and open source software used to build containers, the number of reported vulnerabilities grows sharply with the proliferation of container images and running containers in production. Organizations can easily lose control of security risks and waste developers' time if vulnerabilities are not effectively prioritized and vulnerability management is not integrated into the entire application lifecycle. Here are steps you can take to get control of risk from vulnerabilities:

- Automatically prioritize vulnerabilities by knowing what is used at runtime.
- Embed scanning into CI/CD pipelines and registries to prevent risky images from being deployed.
- Scan for vulnerabilities in both OS and non-OS packages.
- Validate image by checking instructions, user privilege, presence of secrets, and labels.
- Identify new vulnerabilities impacting containers deployed in production.
- Include host scanning (baremetal, VMs, cloud instances) in your vulnerability management.
- Alert the right team for each issue and integrate response within their CI/CD tool.
- By integrating security analysis and compliance validation into this process, you can address issues earlier so you don't slow down deployment. This is known as "shifting security left."

3

# Detect and respond to threats



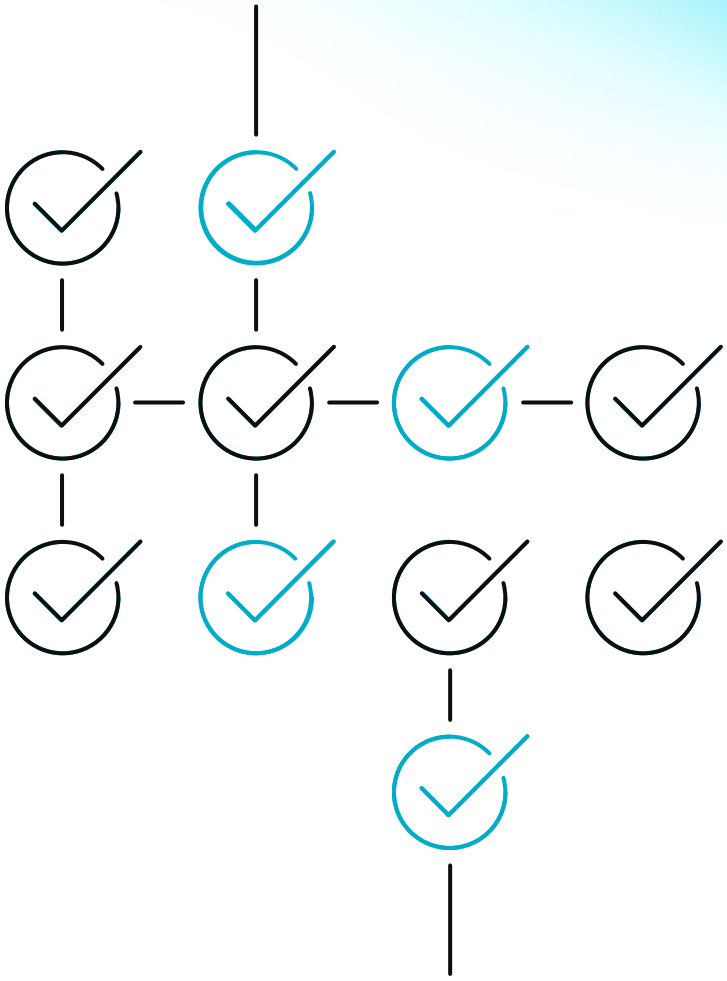
Cybercrime is thriving in the complexity and growing attack surface of cloud-native workloads, cloud services and user permissions. Multilayered threat detection, combining rules-based policies and Machine Learning (ML) is the most effective way to keep up with the evolving threat landscape. Look for context-rich events, automatic actions, and high-fidelity incident data, making sure that you can investigate even after containers are gone.

- Implement unified threat detection for workloads, cloud services, and user activity.
- Use curated policies, if available, to start with strong protection from day one and stay protected against emerging threats.
- Use ML-based detections to protect against highly mutant, polymorphic malware, designed to evade detections, such as cryptominers.
- Block attacks and enforce immutability principle by preventing container drift.
- Monitor your resources, such as CPU, memory, disk, network, and other resource usage as some attacks could be first detected as monitoring alerts rather than security violations.
- Implement Kubernetes network security by applying zero trust and least privilege principles to reduce impact blast radius and lateral movement risk.
- Streamline incident response and quickly respond to threats with context-rich events.
- Use capture files, based on syscall data enriched with Kubernetes and cloud context, to quickly answer the questions of “when,” “what,” “who,” “where,” and “why” for your container security incidents. This detailed record allows you to conduct post mortem analysis and determine root cause, even after containers are gone.

4

# Continuously validate compliance

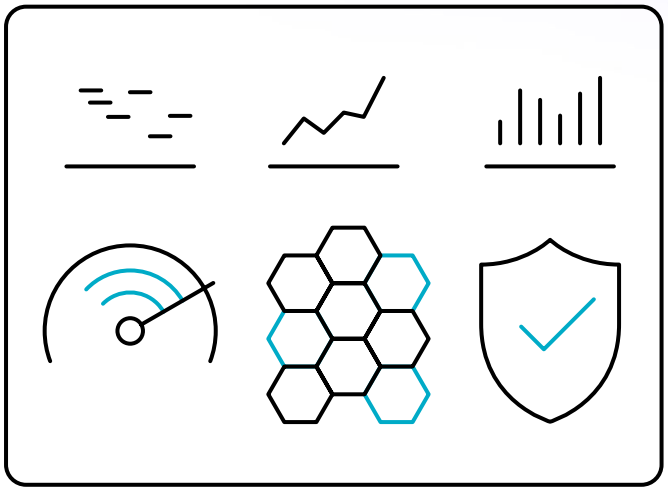
Implement compliance checks to meet regulatory compliance standards (CIS, SOC2, PCI, NIST 800-53, etc.) across containers, Kubernetes, and cloud environments. Monitor cloud services continually for configuration drift that can impact compliance. Measure compliance progress with scheduled assessments and detailed reports.



- Check your cloud control plane, containerized applications and platform configuration against CIS benchmarks and industry best practices.
- Validate compliance during the build, mapping container image scanning policies to standards (e.g., NIST, PCI, SOC2, or HIPAA) or internal compliance policies (e.g., blacklisted images, packages, or licenses).
- Manage compliance at runtime through a rich set of Falco rules for security standards.
- Implement File integrity Monitoring (FIM) to detect tampering of critical system files, directories, and unauthorized changes.
- Enable Automation, eliminate manual processes and enforce compliance with automated remediation, mapping misconfigurations in production to infrastructure as code (IaC) manifests.
- Show proof of cloud and container compliance using cloud audit logs and container forensics data.

5

# Monitor and troubleshoot containers, K8s, and cloud



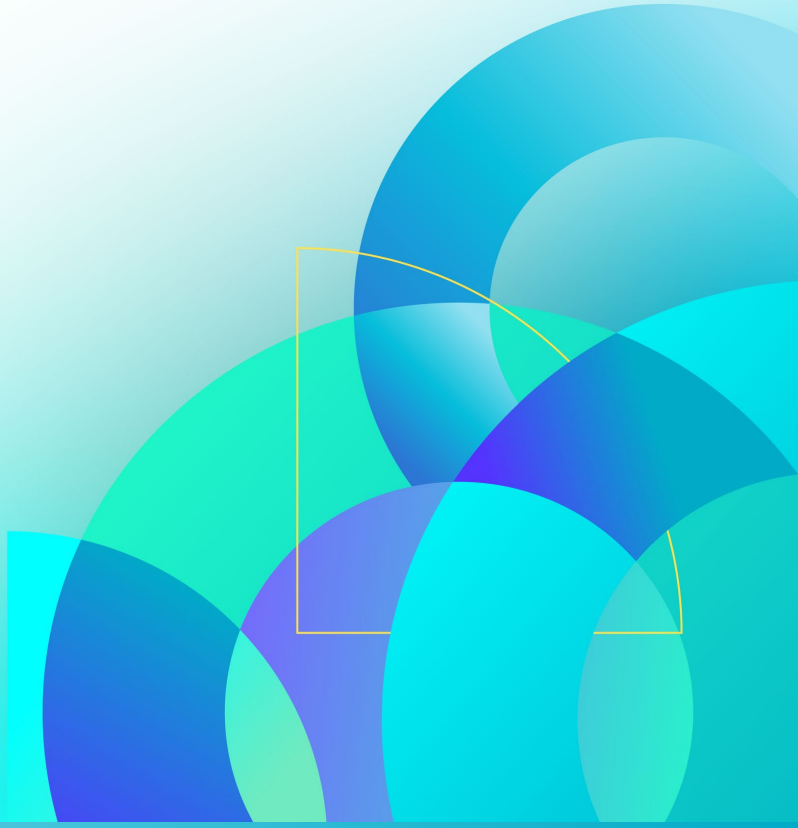
Containers are short-lived, dynamic, and churn constantly. Once a container dies, everything inside is gone. You cannot Secure Shell (SSH) or look at logs, and most of the traditional tools used for monolithic applications don't help when something goes wrong!

- Monitoring the dynamic nature of container-based applications is critical for the high availability and performance of cloud services. Microservices-based applications can be distributed across multiple instances, and containers can move across multi-cloud infrastructure. Monitoring the Kubernetes orchestration state is crucial to understanding if Kubernetes is keeping all of the service instances running.
    - Monitor health and performance with deep visibility into infrastructure, services, and applications. Get the operational status of you r cluster with Kubernetes orchestration monitoring.
    - Immediately identify owners for issue resolution using container and cloud context.
    - Identify pods consuming excessive resources and monitor capacity limits. Control unexpected billing and application rollouts and rollbacks of deployment by monitoring auto-scaling behavior.
    - Reduce cost by optimizing capacity across clusters and cloud.
- Improve application performance and rapidly solve issues with deep container visibility and granular metrics enriched with Kubernetes and cloud context. You can monitor the impact of a given security incident on service availability.
- Get productive quickly by using Promcat.io, a resource catalog of Prometheus integrations with curated, documented, and supported monitoring integrations for Kubernetes platform and cloud-native services.

Sysdig is driving the standard for cloud and container security. The company pioneered cloud-native runtime threat detection and response by creating Falco and Sysdig Open Source as open source standards and key building blocks of the Sysdig platform. With the platform, teams can find and prioritize software vulnerabilities, detect and respond to threats, and manage cloud configurations, permissions, and compliance. From containers and Kubernetes to cloud services, teams get a single view of risk from source to run, with no blind spots, no guesswork, and no wasted time.

**www.sysdig.com**

**REQUEST PERSONALIZED DEMO**



sysdig


CL-001 Rev. G 2/23