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, 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 in your AWS environment to implement an efficient and secure DevOps workflow? With the right set of integrated tools, you can automate security and compliance to manage cloud and container security risks for all your AWS accounts and workloads.

The key is to reduce your risk from cloud misconfigurations, continuously scan for cloud and container vulnerabilities, detect abnormal activity, and prioritize threats to ensure your 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 and securely run containers, Kubernetes, and cloud operations in AWS.

---

1 Gartner: Innovation Insight for Cloud Security Posture Management
2 2021 Sysdig Container Security and Usage Report
Continuous cloud security is required to immediately identify configuration errors and suspicious behavior, and it is the responsibility of AWS users according to the shared responsibility model. The following steps can help you validate your cloud security posture:

1. **Enable Continuous Cloud Security**
   - Automatically discover the assets running in your AWS environment including systems, applications, and services like VPCs, RDS, S3 buckets, ECS, EKS, and Fargate.
   - Check your cloud configurations periodically against Center for Internet Security (CIS) benchmarks to identify misconfigurations (e.g., public storage buckets, exposed security groups and access controls, etc.) and take steps to remediate violations.
   - Detect unexpected changes and suspicious activity across all cloud accounts, users, and services by parsing AWS CloudTrail logs using open-source Falco rules.
   - Use a Cloud Infrastructure Entitlements Management (CIEM) solution to gain visibility into cloud users and services that have access to cloud resources to remediate excessive permissions.
Manage Vulnerabilities: Scan container images and hosts

As the number of container images, versions, and builds proliferates, you lose control of what software is being used and whether software updates are applied. Embedding security into your delivery pipeline as you build applications helps you identify and address vulnerabilities faster, and keeps your developers productive. You can start with scanning your Elastic Container Registry (ECR) in AWS, and then take these steps to maintain continuous control:

- Embed scanning into CI/CD pipelines like AWS CodePipeline and registries to prevent risky images from being deployed.
- Adopt in-line scanning and maintain full control of your images.
- Automate image scanning for short-lived Fargate tasks to reduce risk of running vulnerable containers.
- Validate the build configuration (Dockerfile instructions) and image attributes (like size and labels).
- Identify new vulnerabilities that impact the image once the container has been deployed.
- Create different policies for each workflow, including images from public repositories and images built in-house. Consider different checks for each app.
- Alert the right team for each issue (notify the owner of each image and integrate with your CI/CD tool to show the scan results directly in that context).
- Automatically scan Amazon EC2 virtual and bare metal host instances to identify vulnerabilities in OS and non-OS packages.

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.”
You can reduce runtime risk by configuring applications with minimum privilege and access permissions. Your policies should also monitor for anomalous behavior and configuration drift. Creating policies that can prevent attacks without breaking the applications is challenging. Be sure to capture a detailed record for incident response. Consider these steps for reducing your runtime risk.

Leverage Kubernetes-native controls available with EKS for runtime protection of cloud-native workloads.

→ Use Admission Controllers to allow or block specific configurations and determine whether the container can be run on the cluster.
→ Prevent attacks enforcing "least privilege" on containers through Pod Security Policy (PSP). PSPs control what permissions pods get at runtime (e.g., which user is running in privileged mode, whether they have access to the host network or filesystem, etc.).

Monitor CPU and other resource usage as they are typically exploited in DoS and crypto-mining attacks. Monitoring network connections gives you information about the attack, runtime behavior, and spread vectors. Some attacks are first detected as monitoring alerts rather than security violations.

Create and maintain runtime security policies that observe workload behavior, monitor cloud activity, and identify anomalous events for ECS, EKS, and Fargate.

→ Leverage tools to automatically build and customize policies or use out-of-the-box Falco rules.
→ Implement least-privilege and compliant network policies with Kubernetes and app metadata.
→ Simulate the effects of runtime policies before applying them in production to avoid breaking application functionality.
→ Visualize network communication in and out of a particular pod/service/app/tag over time by using topology maps.
→ Apply the right security policy based on container role and Kubernetes context.
→ Automate use of events in AWS CloudTrail logs to detect threats and configuration changes on cloud services.

Streamline incident response and quickly respond to container and cloud security threats with a detailed activity record in ECS, EKS, and Fargate. Use capture files based on syscall data to quickly answer the questions of “when,” “what,” “who,” 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.
Implement compliance checks to meet regulatory compliance standards (CIS, SOC2, PCI, NIST 800-53, etc.) across containers, Kubernetes, and cloud. Check regularly against CIS Security best practice benchmarks. Monitor cloud services continually for configuration drift that can impact compliance. Measure compliance progress with scheduled assessments and detailed reports.

**Continuously validate compliance**

Check your container and platform configuration against CIS benchmarks for AWS, Docker, and Kubernetes.

Validate compliance during build-time by 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).

Implement File integrity Monitoring (FIM) to detect tampering of critical system files, directories, and unauthorized changes. FIM is a core regulatory requirement for a number of compliance standards.

Manage compliance at runtime. Check for best practices (e.g., don’t run privileged containers and don’t run containers as root) and look for known adversary tactics and techniques. Achieve and maintain compliance with security frameworks mapping through a rich set of Falco rules for security standards and benchmarks, like NIST 800-53, PCI DSS, SOC 2, MITRE ATT&CK®, CIS AWS, and AWS Foundational Security Best Practices.

Provide proof of compliance with capture files that incorporate detailed forensics data and host scanning reports. It’s important to record configuration and policy changes, including an audit of runtime changes for compliance audits.
Monitor and troubleshoot containers, Kubernetes and cloud

Containers and cloud services are dynamic and churn constantly. Visibility into the health and performance of your AWS workloads and infrastructure is critical for ensuring the availability of your cloud applications.

Implement monitoring built for cloud native infrastructure, applications, and AWS services. Microservices can be distributed across multiple instances, and containers can run across different regions and multi-cloud infrastructure. To improve application performance and rapidly solve issues, you need deep container, infrastructure, and service visibility and granular metrics enriched with Kubernetes and cloud context.

- Immediately identify owners for issue resolution using container and cloud context.
- Identify pods consuming excessive resources and monitor capacity limits.

Tap into open-source Prometheus for monitoring AWS services and cloud-native applications. Extract Prometheus metrics via AWS CloudWatch for services like Fargate and S3, and view with dashboards in Grafana.

- 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.

Monitor the Kubernetes orchestration state to understand if Kubernetes is healthy and providing the right resources to keep all of your service instances running. Visualization of key metrics about the state of your nodes, namespaces, pods, etc., will help you identify if you have sufficient resources.

- Control unexpected billing and application rollouts and rollbacks of deployment by monitoring auto-scaling behavior.
- Reduce cost by optimizing capacity across clusters and clouds.

Capture and store data to ensure you can investigate and solve issues quickly. Once a container dies, everything inside is gone. You can't shell into a stopped container to see what happened! Audit logs and detailed activity information will help you successfully determine root cause, even for containers that are no longer running.
Confidently secure containers, Kubernetes, and cloud with the Sysdig Secure DevOps Platform. Scan images, detect and respond to threats, validate cloud posture and compliance, monitor, and troubleshoot.

www.sysdig.com

START YOUR FREE TRIAL

REQUEST PERSONALIZED DEMO