sysdig

The Grand Atlas of Software Security

What you need at every stage of the pipeline

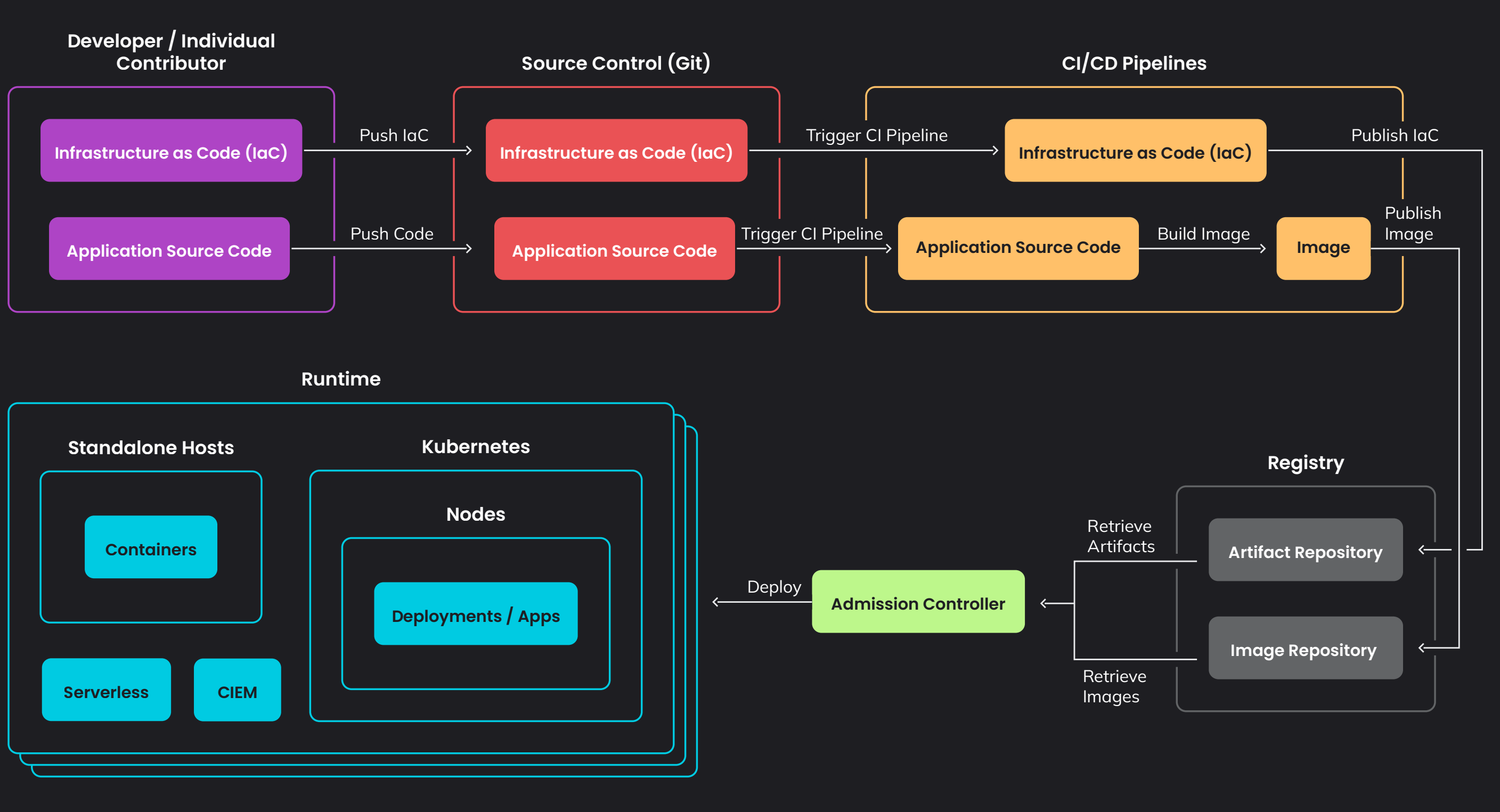
This infographic demonstrates how to secure each stage of the software lifecycle, with a focus on the **Shift Left** approach, where early remediation reduces risks and costs. Sysdig integrates security checks throughout the development pipeline, enabling developers to catch and fix issues early using tools like VSCode, Jenkins, or GitHub Actions. By integrating security feedback into developer environments and ticketing systems like Jira or GitHub Issues, Sysdig ensures vulnerabilities are addressed promptly and efficiently.

Key security concepts integrated into this process include defense in depth, least privilege, zero trust, security by default, and continuous compliance.

- **Defense in depth:** Multiple layers of security controls ensure vulnerabilities are caught at various stages. Even if an issue slips through early, it's likely to be detected later, such as in the CI/CD pipeline or runtime.
- **Least privilege:** This principle ensures users and services have only the access they need, minimizing risks. Sysdig helps identify and fix over-permissioned roles or configurations to reduce the attack surface.
- **Zero trust:** Sysdig continuously verifies components at every stage, ensuring only trusted elements make it to production. Even during runtime, it watches for and flags any anomalies that arise.
- **Security by default:** By embedding secure practices early on, Sysdig reduces the chance of misconfigurations. This ensures security is baked into the development process, not added as an afterthought.
- **Continuous compliance:** Regular automated scans ensure ongoing adherence to security policies, catching new vulnerabilities as they emerge and preventing security drift.

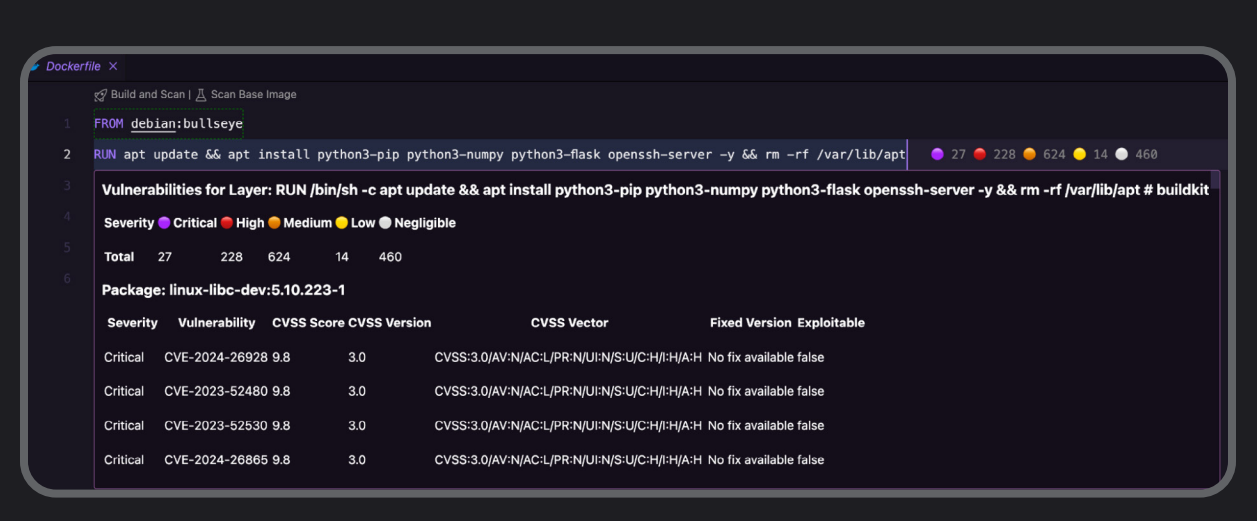
Sysdig also leverages **immutable infrastructure** and **declarative IaC** to maintain consistency across environments, reducing configuration drift and improving traceability. Security checks are frictonless, smoothly integrating with DevOps workflows without slowing development in fast-paced environments like microservices and cloud.

Finally, **runtime threat detection** identifies zero-day vulnerabilities and suspicious behaviors in running workloads, ensuring strong last-line protection against active threats.



Developer / Individual Contributor

Individuals who write, test, and maintain codebases. They utilize tools and environments to develop **infrastructure as code (IaC)** and **application source code.**



With Sysdig, you can scan for misconfigurations and vulnerabilities while developing by using our **Visual Studio Code extension** or running the **Sysdig CLI Scanner.**

Infrastructure as Code

When developing infrastructure as code using tools like **Terraform** or **Helm charts**, it's essential to adhere to compliance standards such as the CIS Benchmarks and NIST SP 800-53 to ensure security and regulatory compliance.

Developers should enforce best practices like least privilege principles, proper secrets management, and regular code reviews to maintain a strong security posture.

Application Source Code

Developers should actively look for vulnerabilities in their code and the libraries or packages they use. Catching issues early allows them to fix problems swiftly and efficiently, preventing minor flaws from becoming major security risks in the future.

Source Control (Git)

Manages code repositories, enabling versioning, branching, merging, and collaboration among developers.

Enable a **Git Integration** within Sysdig to automatically detect misconfigurations in your repositories, and open Pull Requests with recommended patches from Sysdig.

Infrastructure as Code

Remediations applied at this stage make developers aware of patches, speeding up resolution. Integrating security checks here ensures that issues are caught in the development process, when they are easiest to fix. Early detection also fosters a security-first mindset among developers, promoting better coding practices.

Application Source Code

Securing applications from the source code at Git repositories is key to catching vulnerabilities early. It ensures issues are fixed before affecting pipelines or runtime. Automated checks in Git workflows help maintain security without slowing development.

CI/CD Pipelines

Continuous integration and continuous deployment pipelines automate the process of integrating code changes, running tests, and deploying applications.

remote-scan-from-registry summary

Scan Results for sysdiglabs/dummy-vuln-app:latest

	Critical	High	Medium	Low	Negligible
Total Vulnerabilities	4	12	9	3	0
Fixable Vulnerabilities	3	11	9	3	0

GitHub, GitLab, Jenkins, and **Azure Pipelines** are just a few examples of where you can scan your CI/CD pipelines with Sysdig.

Infrastructure as Code

Security checks in CI/CD pipelines catch any lingering IaC misconfigurations before deployment, preventing insecure infrastructure setups from reaching production. Automated scanning in the pipeline ensures consistent enforcement of security policies across different environments.

Application Source Code

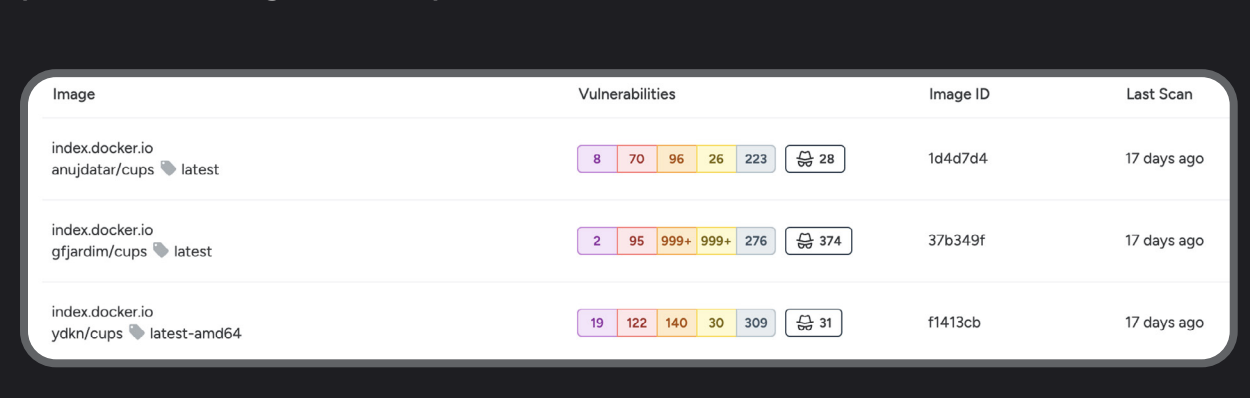
At this point the application gets built, and if there's no version pinning for dependencies, new security concerns arise as it fails for reproducibility and may introduce new vulnerabilities that didn't exist at development. Usually, applications are packaged in container images.

Image

Now it's not only a matter of application vulnerabilities, but also bad practices at the container image building instructions. This can be OS-level dependencies, exposed secrets, bad default settings as root users, or any further security breach inherited from the base image. Pipelines should fail if images don't meet security policies, stopping them from reaching the public registry.

Registry

A centralized storage system for build artifacts and container images, facilitating easy retrieval and deployment. Keep in mind that at this stage, a single vulnerability or misconfiguration can lead to thousands or even millions of affected instances upon deployment. This is why securing the previous stages is important.



Utilize the **Sysdig Registry Scanner** chart to scan your registries across a comprehensive list of supported vendors.

Artifact Repository

Artifacts in the repository must pass security checks before they are stored, ensuring only secure components are available. Regular scans catch newly discovered vulnerabilities. This helps maintain a safe and compliant repository for future builds.

Image Repository

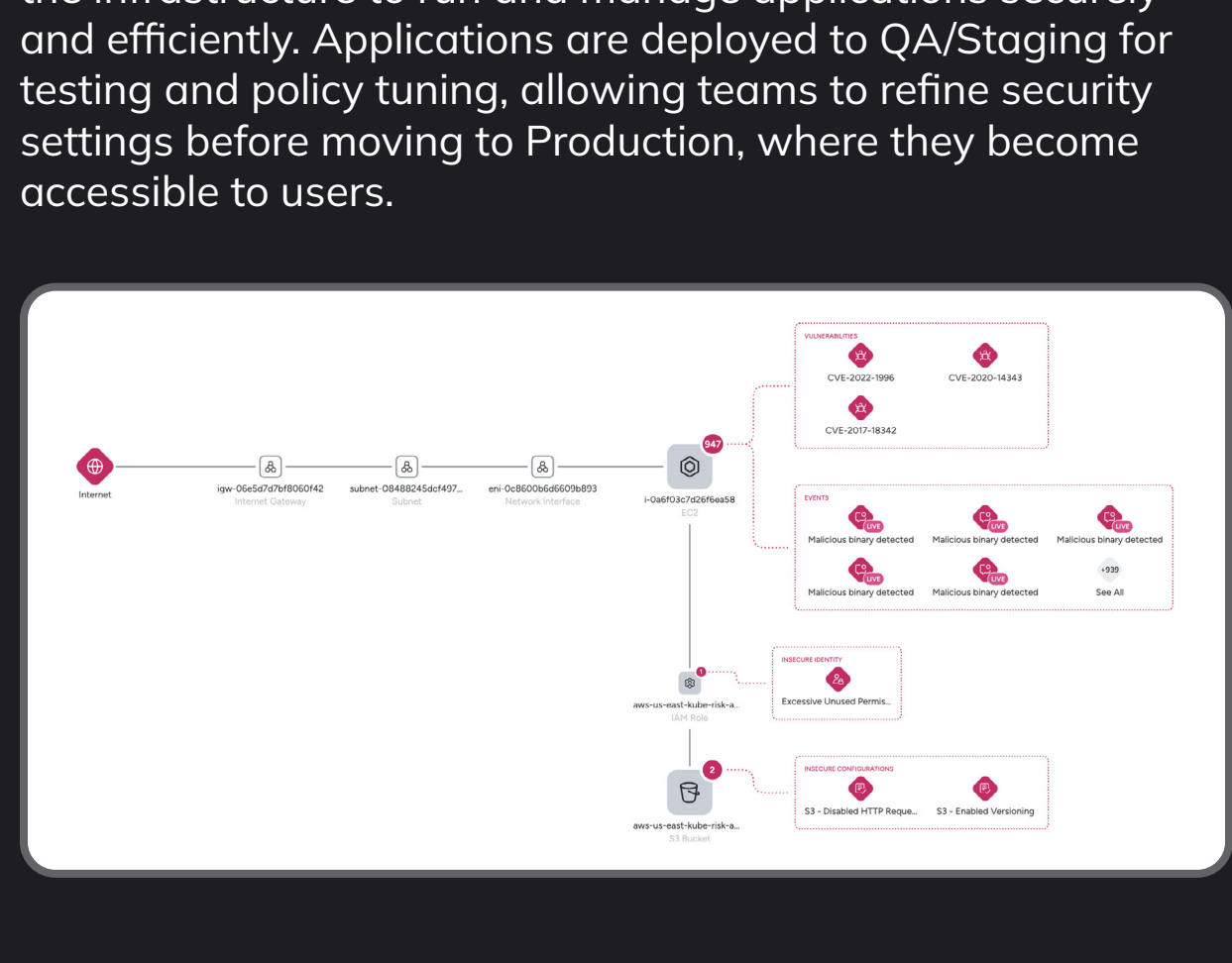
Images should undergo continuous scanning to detect vulnerabilities that emerge after they're stored. Only trusted, secure images should be allowed for deployment. Regular rescans ensure that even stored images meet up-to-date security standards.

Admission Controller

The admission controller acts as the last security checkpoint before deployment, ensuring only compliant workloads are allowed to run. It verifies that both infrastructure and application components meet predefined security policies. If issues are detected, it can block deployment or flag them for review. This final layer helps catch anything missed in earlier stages, reinforcing a **zero-trust** approach. Use the **Sysdig Admission Controller** to only allow deployments that pass specified policies.

Runtime

Runtime environments host applications in the cloud or on-premises, including **standalone hosts, Kubernetes clusters, serverless platforms,** and **CIEM systems.** They provide the infrastructure to run and manage applications securely and efficiently. Applications are deployed to QA/Staging for testing and policy tuning, allowing teams to refine security settings before moving to Production, where they become accessible to users.



Sysdig is capable of correlating everything at your **Posture, Vulnerabilities**, or **Runtime Threats** for all your assets at your Inventory.

Standalone Hosts

At runtime, standalone hosts must be continuously monitored for vulnerabilities and suspicious activity. Regular security checks help ensure that hosts are not running outdated or insecure configurations. **Sysdig Agents** running in the host identify all suspicious activity at these containers.

Containers

These should be continuously monitored for vulnerabilities during runtime, and security needs to act quick as these may only live seconds. Sysdig's **Container Drift** detects and prevents any external binary from being executed.

Kubernetes

Kubernetes security focuses on protecting workloads by managing vulnerabilities in containers and their configurations. Proper resource limits and role-based access control (RBAC) are essential to reduce exposure. Incidents are discovered by analyzing the **Kubernetes Audit Logs.**

Nodes

Like standalone hosts, Kubernetes deploys a Sysdig Agent on each node (via DaemonSet), detecting all workload unexpected activity and preventing malicious actors from executing malware. With **Captures** and **Sysdig Inspect**, you can analyze incidents in granular detail.

Containers

Reinforce strong container networking using the Network Policies generated by Sysdig.

Serverless

Serverless environments require strict security policies to prevent unauthorized access and misconfigurations. Enforcing compliance ensures only approved functions are deployed and executed. Function isolation is critical to reducing the attack surface and limiting potential breaches.

CIEM

CIEM is crucial for security as it helps manage and minimize excessive permissions, reducing the risk of unauthorized access to critical cloud resources. By enforcing least privilege, CIEM ensures that users and services only have the access they truly need, lowering the chance of privilege escalation attacks. Sysdig assists in CIEM by continuously analyzing cloud permissions, identifying misconfigurations or overly permissive roles, and offering actionable insights to reduce risk.

sysdig

INFOGRAPHIC

ING-011 REV. A 11/24

About Sysdig

In the cloud, every second counts. Sysdig stops cloud attacks in real time by instantly detecting changes in risk with runtime insights and open source Falco. We correlate signals across workloads, identities, and services to uncover hidden attack paths and prioritize the risks that matter most.