# Sysdig Secure Architecture Guide

sysdig

Sysdig's mission is to make every cloud deployment secure and reliable. We were the first company to offer deep visibility with rich context for both public cloud and on-premises container and Kubernetes environments. We maintain our leadership position in cloud and container security by providing an unparalleled source-to-run experience for modern applications.

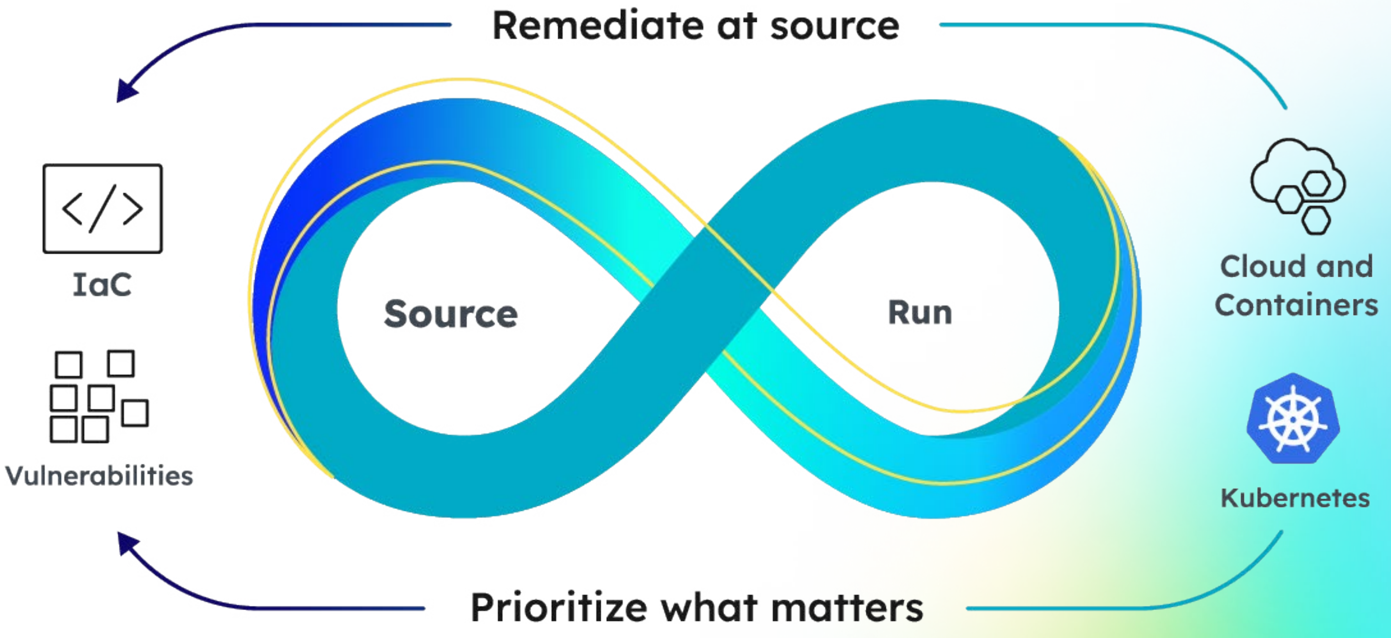
Sysdig Secure addresses the dynamic and evolving requirements of secure software delivery and runtime security operations from the first code commit to enterprise production scale. Sysdig Secure is designed to meet the needs of small and large teams of any maturity level through a combination of friendly onboarding, heavily curated security content, guided remediation, automation, and extensive customization options.

## Commitment to Open Source Software

Sysdig was founded on open source tools that provide deep visibility for security and monitoring. Today, Sysdig's capabilities cover the entire spectrum for securing and monitoring modern applications from source to run.

Falco is the rule language that underpins the Sysdig Secure product. Falco was the second major open source project that the Sysdig engineering team created, which we then contributed to the Cloud Native Computing Foundation in 2018. We remain an active supporter of the Falco project, including contributing to its extensibility to accommodate cloud log detections from custom sources.

We are committed to a single rule and policy engine across security features, enabled by Falco and Open Policy Agent (OPA), respectively. We also leverage eBPF to instrument the Linux kernel to collect system call data.

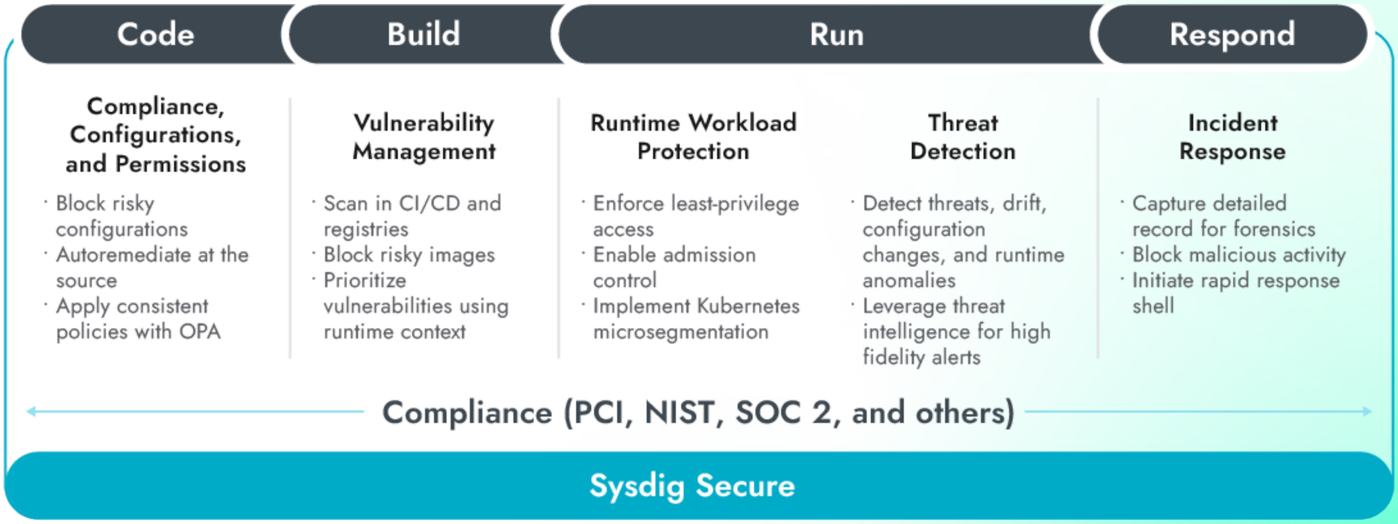


# Table of Contents

01

# Abstract

This architecture guide is intended to help security professionals understand Sysdig's capabilities, integration points, and overall design as they consider how this solution fits into their unique IT environments. Each section takes a use-case-centric approach to detail a set of relevant features, and express how those features connect to each other and to the source-to-run security strategy.



Code | Build | Run | Respond

**Compliance, Configurations, and Permissions**

- Block risky configurations
- Autoremediate at the source
- Apply consistent policies with OPA

**Vulnerability Management**

- Scan in CI/CD and registries
- Block risky images
- Prioritize vulnerabilities using runtime context

**Runtime Workload Protection**

- Enforce least-privilege access
- Enable admission control
- Implement Kubernetes microsegmentation

**Threat Detection**

- Detect threats, drift, configuration changes, and runtime anomalies
- Leverage threat intelligence for high fidelity alerts

**Incident Response**

- Capture detailed record for forensics
- Block malicious activity
- Initiate rapid response shell

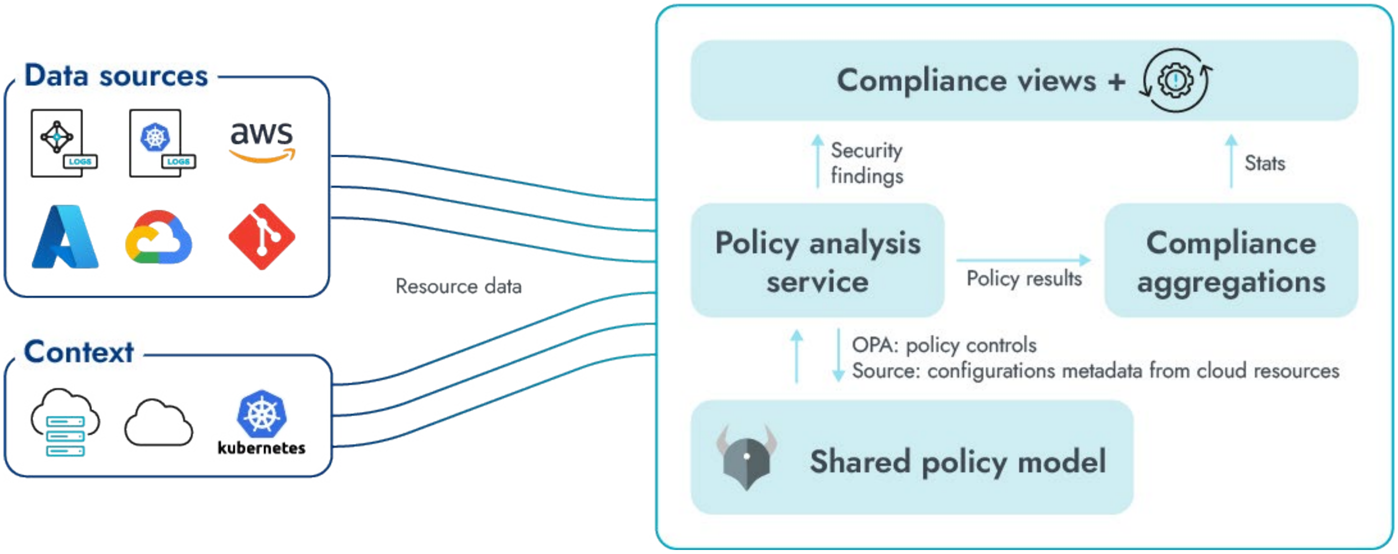
Compliance (PCI, NIST, SOC 2, and others)

Sysdig Secure

02

# Compliance, Configurations, and Permissions (CSPM)

Sysdig Cloud Security Posture Management (CSPM) continuously manages cloud infrastructure and identity risks by identifying and enabling the remediation of misconfigurations in the cloud control plane, cloud resources, and cloud-deployed workloads. CSPM applies common frameworks, regulatory requirements, and internal company policies to proactively and reactively assess target environments against security standards.



Sysdig CSPM can check cloud and Kubernetes environments against all major compliance standards out of the box, including PCI DSS, NIST 800-53, CIS, SOC2, GDPR, and many more. CSPM explicitly maps out-of-the-box policies onto the relevant compliance frameworks, which are weighted as high, medium, or low. Users see an overall compliance score based on the weighted average, rather than a generic pass/fail result. The weighted approach accounts for the fact that not all compliance requirements are equal sources of risk, and allows users to make more granular risk-based decisions about their environments.

# Configuration Assessment

Sysdig CSPM starts with asset discovery in the cloud environment, including infrastructure-as-a-service (IaaS) instances, Kubernetes clusters, containers, identities, policies, and so on. CSPM maintains assets in an inventory database and evaluates their configurations against multiple policies. Policies can be based on regulatory and compliance standards such as CIS Benchmarks, NIST, and PCI DSS, or customized to suit an organization's internal requirements.

Sysdig CSPM is an agentless solution that runs in Sysdig's software-as-a-service (SaaS) back end and uses API-based queries to access environments. CSPM fully supports AWS, Azure, and GCP.

## Policy-as-Code (PaC)

Sysdig CSPM policies are a collection of requirements, which map to controls. An example of a policy might be PCI DSS, if users needed to comply with that standard. A **requirement** might be a specific element of PCI DSS, such as "2.1 – Change vendor defaults." PCI DSS 2.1 indicates to "always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network."

A requirement contains one or more **controls**. A control is a specific check for a given resource type evaluating a particular configuration and its associated remediation playbook. A control can satisfy any number of requirements. Controls are assigned a severity based on the potential impact if an attacker were to leverage the misconfiguration.

Sysdig CSPM is an agentless solution that runs in Sysdig's SaaS back end.

Requirements, when assessed in an environment, will either pass or fail. The requirement will pass if all controls pass for all evaluated resources, and fail otherwise. If a requirement fails, CSPM will display the severity of failing controls.

CSPM provides out-of-the-box mappings of controls to requirements for these compliance standards, adding new policies regularly based on customer demand:

- CIS Benchmark for Kubernetes, Docker, and Linux
- CIS Foundations for AWS, GCP, and Azure
- AWS Best Practices Compliance Framework
- AWS Well Architected Compliance Framework
- PCI DSS
- NIST
- ISO 27001-2013
- SOC
- FedRamp
- GDPR
- HIPAA
- HiTrust CSF

Through a custom policy wizard, users can create new Rego-based policies and controls, generating PaC, without having to write a single line of code.

Sysdig's shared policy model (SPM) describes all configuration and compliance policy types in a single OPA-based format. The SPM enables the evaluation of a control at any stage from source to run, regardless of the asset's defined configuration.

## Infrastructure-as-Code (IaC)

IaC is a declarative approach to infrastructure management that relies on version-controlled manifest files to define and configure assets and workloads in a repeatable manner. Sysdig maps assets or resources to the IaC manifest files that create or configure them in order to enable drift detection and remediation of violations in the environment. Supported IaC formats include Terraform, Helm, Kustomize, and YAML.

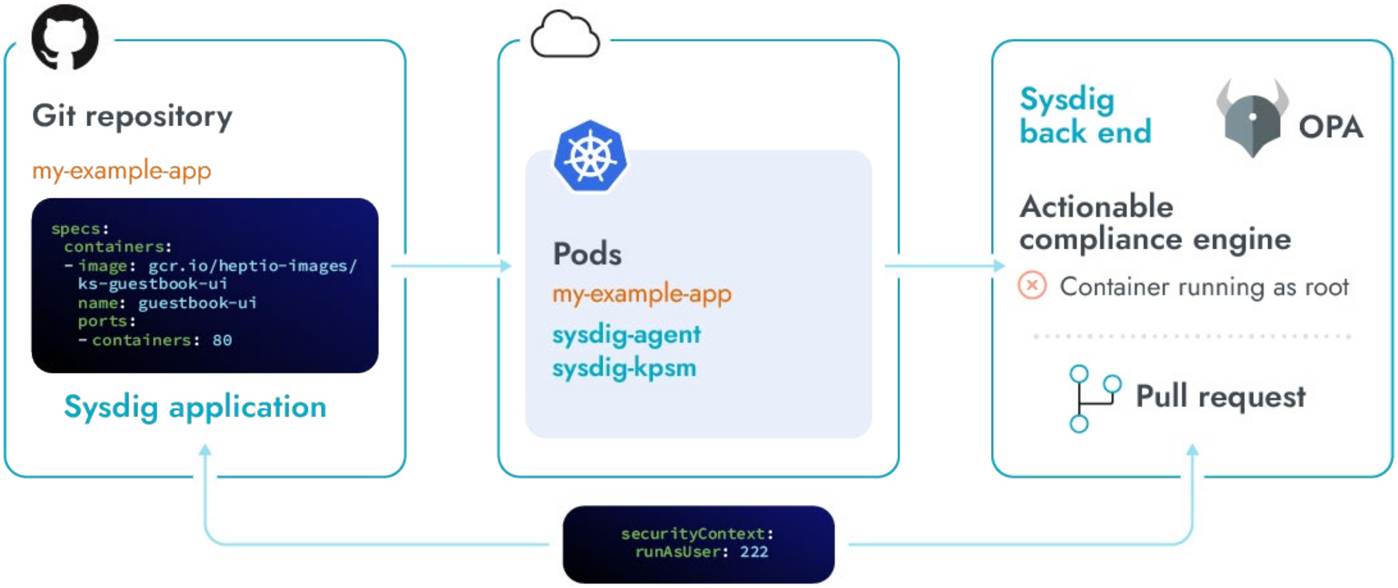
## Policy Enforcement

It's important to evaluate IaC continuously throughout the CI/CD pipeline and workload life cycle, because unlike application code, the workload configuration is likely to change as it proceeds along the pipeline.

There are several stages where policy evaluation and enforcement can occur. The leftmost enforcement point is during the creation of a pull request (PR) that includes an IaC manifest, which triggers a security scan. The merge is blockable if the PR fails the defined policy checks. Enforcement points along the CI/CD pipeline use the Sysdig CLI to identify violations in the IaC artifacts, which can inform the decision of whether to move the workload forward in the pipeline. Sysdig CSPM evaluates exactly the same policy, regardless of when the assessment occurs.

## Remediation

Sysdig CSPM supports remediation both at the source and in run time. Whenever possible, a violation generates a tailored remediation suggestion. If the affected resource has associated IaC artifacts, it is possible to automate the remediation through a PR. For running workloads that do not have associated IaC manifests, Sysdig generates the fix to resolve the specific violations found in run time.



```
specs:
  containers:
  - image: gcr.io/heptio-images/
    ks-guestbook-ui
    name: guestbook-ui
    ports:
    - containers: 80
```

```
securityContext:
  runAsUser: 222
```

### Continuous Compliance and Threat Detection

Compliance scans occur periodically — once per day by default — or customized higher frequencies. Additionally, Falco's cloud connector provides continuous monitoring of the security posture, assessing cloud log activity in near-real time and alerting users about relevant configuration events that occur outside the scanning window.

### Kubernetes Security Posture Management (KSPM)

KSPM is CSPM for Kubernetes environments. In addition to the Sysdig DaemonSet, which analyzes the host configuration, another deployment is required to analyze the configuration of Kubernetes resources. A static scan occurs once per day, in addition to the ongoing monitoring of Kubernetes audit logs.

The Kubernetes admission controller adds another enforcement point, which allows the CSPM system to evaluate containerized workloads before deploying them to the cluster and sends alerts about or blocks requests in the event of a policy violation.

## Permissions Management

Over-permissioned identity is the most common cloud service security misconfiguration. Implementing least privilege is a crucial best practice to help avoid or mitigate risks of data breaches and to contain privilege escalation and lateral movement.

Due to the fine granularity of permissions available in cloud environments, cloud infrastructure entitlement management (CIEM) security is key in applying the least-privileges concept. Carefully giving users exactly what they need to perform their duties is fundamental to avoiding an attacker escalating privileges inside the environment.

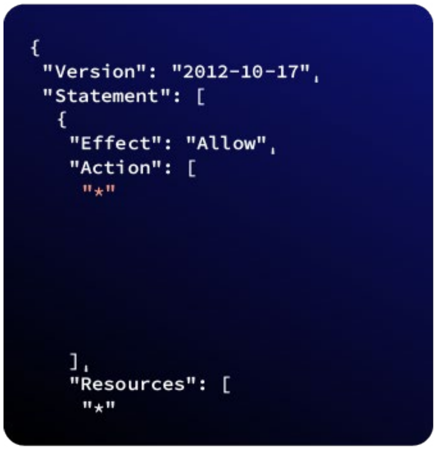
Sysdig Secure analyzes the audit logs of all executed cloud commands in an organization's accounts and correlates them with policies, roles, and users. Under the Identity and Access feature in the Sysdig user interface (UI), a general dashboard informs users of:

- The total permissions given and used.
- How many users are inactive, and which users to consider deleting.
- Averages of permissions per policy and policies per user.
- The policies, users, and roles with the worst cases of unused permissions.

Implementing least privilege is a crucial best practice to help avoid or mitigate risks of data breaches and to contain privilege escalation and lateral movement.

The analysis of unused vs. given permissions is translated into policy suggestions that limit the permissions granted to only what is needed. Offered in JavaScript Object Notation format, it is possible to paste these policy suggestions directly into AWS identity and access management. Using these policy suggestions is a quick and effective way to reduce excessive permissions and achieve the concept of least privilege in AWS environments.

**Current policy**

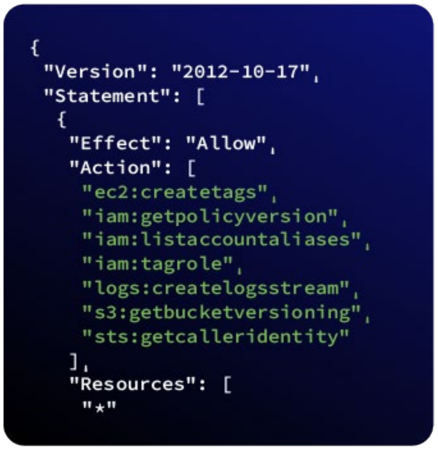


```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "*"



   ],
   "Resources": [
    "*"
```

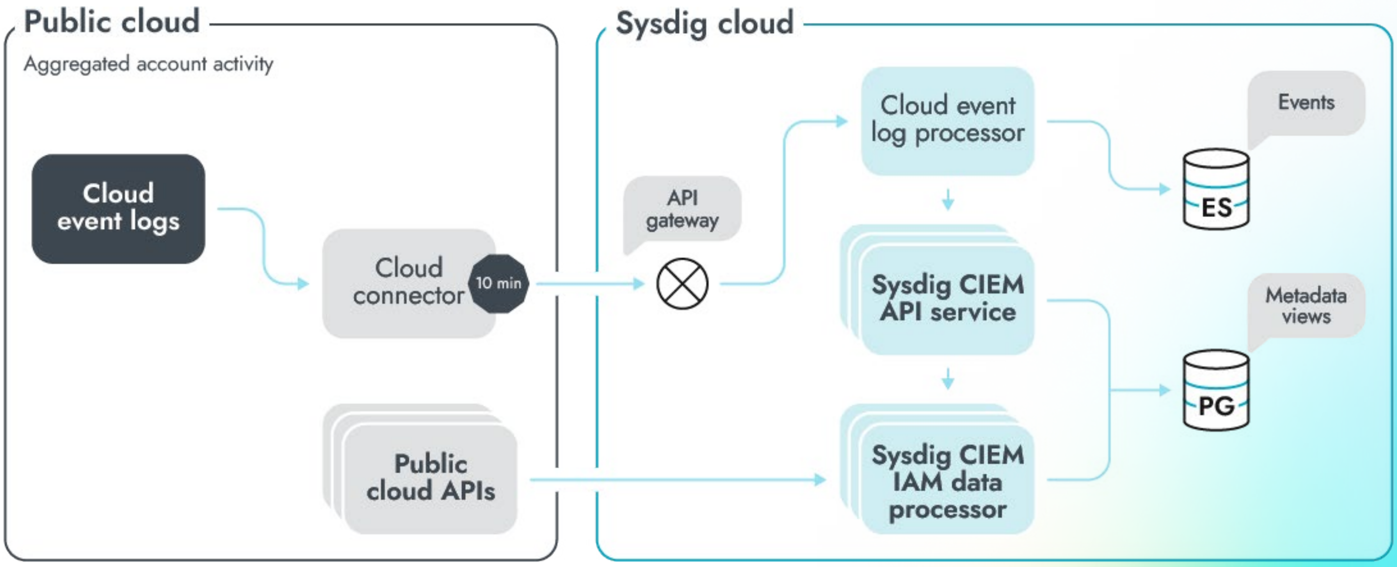
**Suggested policy**



```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:createtags",
    "iam:getpolicyversion",
    "iam:listaccountaliases",
    "iam:tagrole",
    "logs:createlogsstream",
    "s3:getbucketversioning",
    "sts:getcalleridentity"
   ],
   "Resources": [
    "*"
```

Sysdig also highlights risks that are not related to permissions. Users have risk labels to indicate whether they are a root user, do not have multifactor authentication enabled, have not rotated their access keys, or other relevant indicators. Similarly, resources such as S3 buckets will have labels indicating whether they are public, do not have encryption, are cross-account, and so on. These labels help highlight which users are most at risk and which ones to prioritize.

Sysdig Secure back end operates in AWS and GCP and consists of multiple microservices. Every component of the entire back end system is a collection of loosely coupled services collaborating with each other. The CIEM service is one of those services, commonly known as cloudsec.



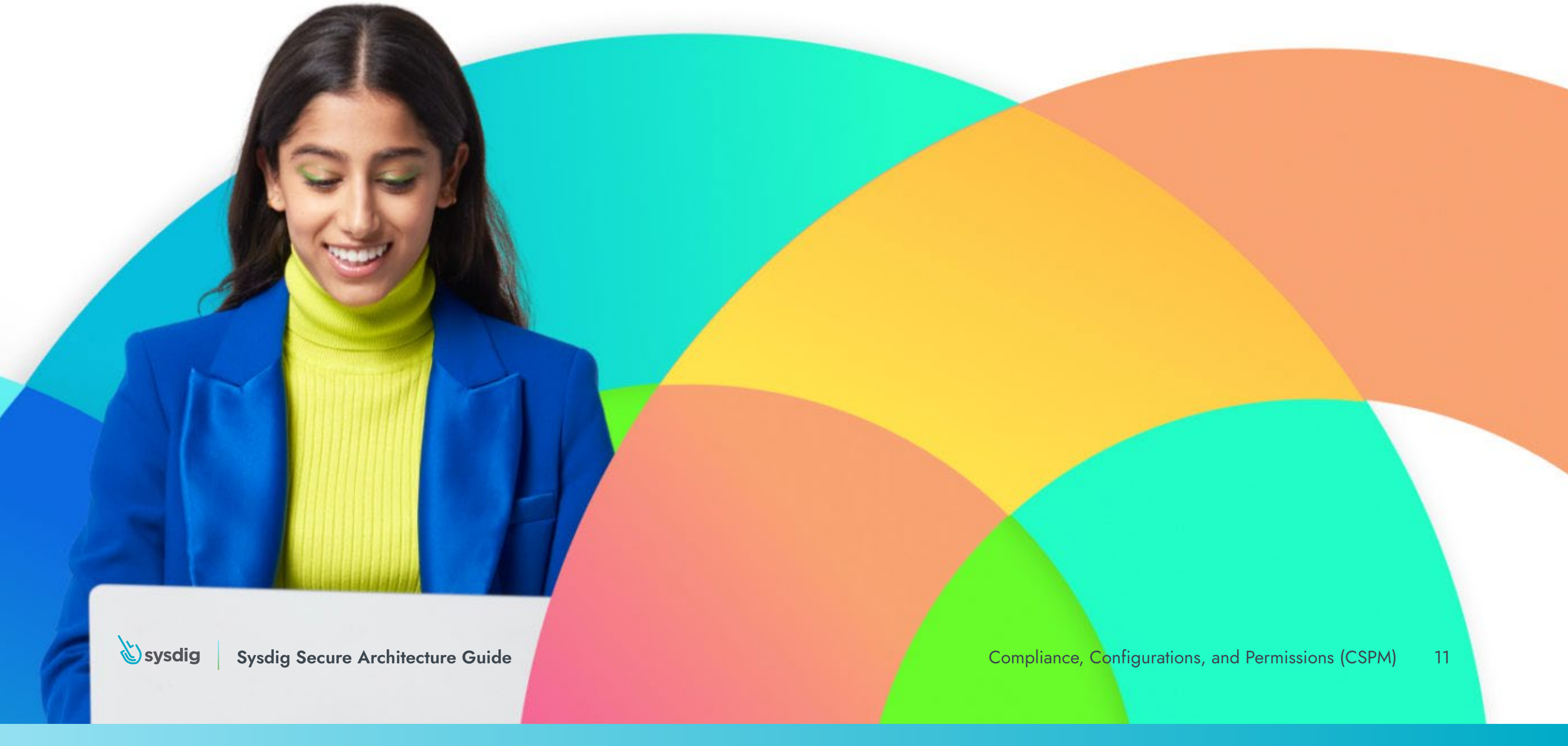
## cloudsec-server

The server is the core component of the cloudsec system that schedules analytics workloads periodically, distributes the tasks among workers, and keeps track of the state of the workload. Analytics workloads are scheduled to run four times a day (every six hours), instead of at one defined time. External applications and microservices within the secure back-end ecosystem interact with cloudsec-server through representational state transfer (REST) APIs. The majority of REST endpoints are primarily used to populate the analytics page, which visualizes the cloud security posture of an organization's cloud accounts. The server also tracks the overall progress of a task and updates its status as active, succeeded, or failed.

## cloudsec-worker

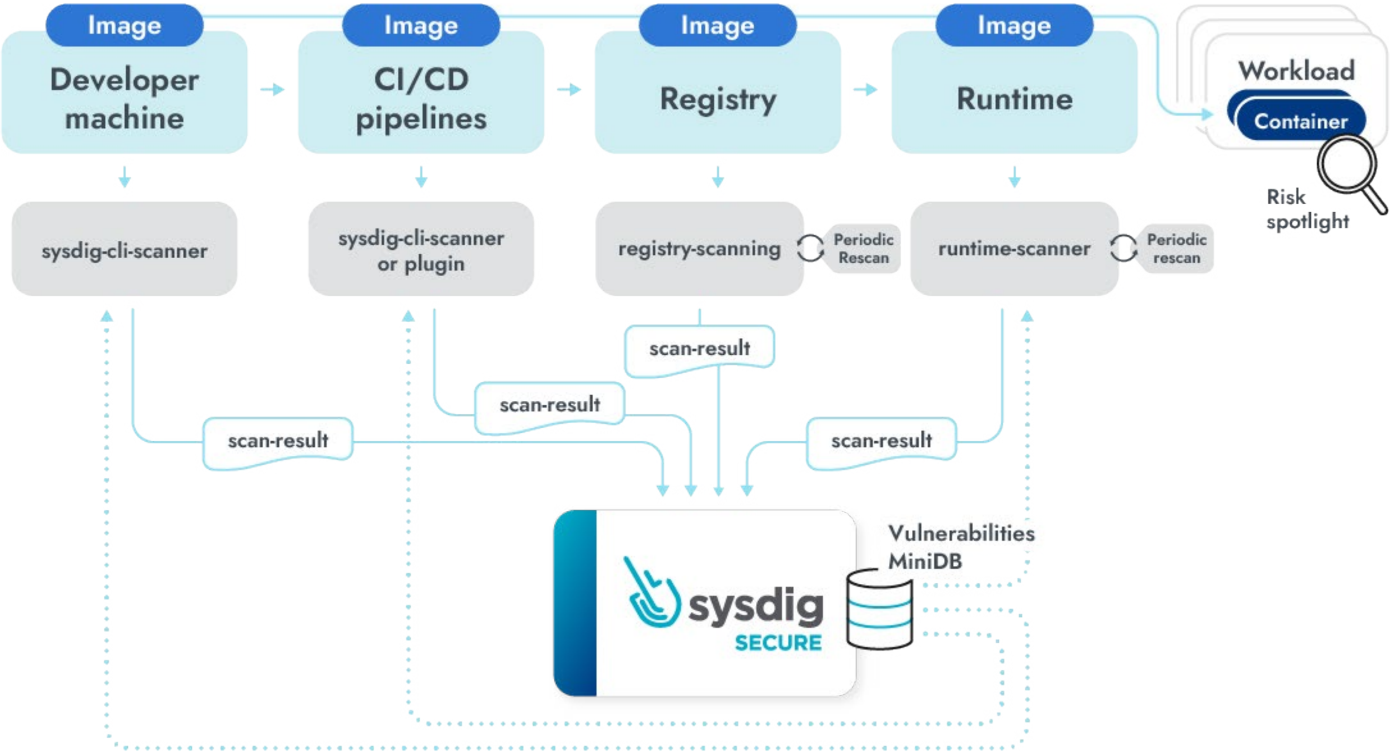
Workers are responsible for running the actual analytics workload. Workers collect metadata associated with those cloud accounts necessary for running analytics tasks. Upon successful completion of metadata collections, the worker performs the necessary computation and reports the status back to the master. The security posture visualization is immediately available on the web interface upon completion of the analytics workload. The worker also marks the intermediate task states, and persists those states to the metadata store.

Sysdig Secure back end operates in AWS and GCP and consists of multiple microservices. Every component of the entire back end system is a collection of loosely coupled services collaborating with each other.



03

# Vulnerability Management



Cloud-native vulnerability management (VM) protects workloads throughout their entire life cycle by enabling both shift-left best practices (via local machine and CI/CD-integrated scanning) and continuous monitoring of the workloads' vulnerability state in production. The former enables more efficient development of secure software, while the latter helps protect against zero-day events and other post-release problems.

Even though users can adapt the scanning components to be distributed along the different platforms and phases in the pipeline, centralized control over the VM security posture and governance remains. Policies are configured centrally, and scan results and reports are produced centrally as well.

# Vulnerability Scanning Engine

Sysdig's scanner architecture uses a distributed design for unparalleled accuracy and performance. The client-side component downloads the vulnerability database, retrieves the policies, downloads the profiling information, and locally extracts the software bill of materials. It then performs vulnerability matching, enriches the vulnerability list with profiling information, evaluates the policies, and sends the result to the back end.

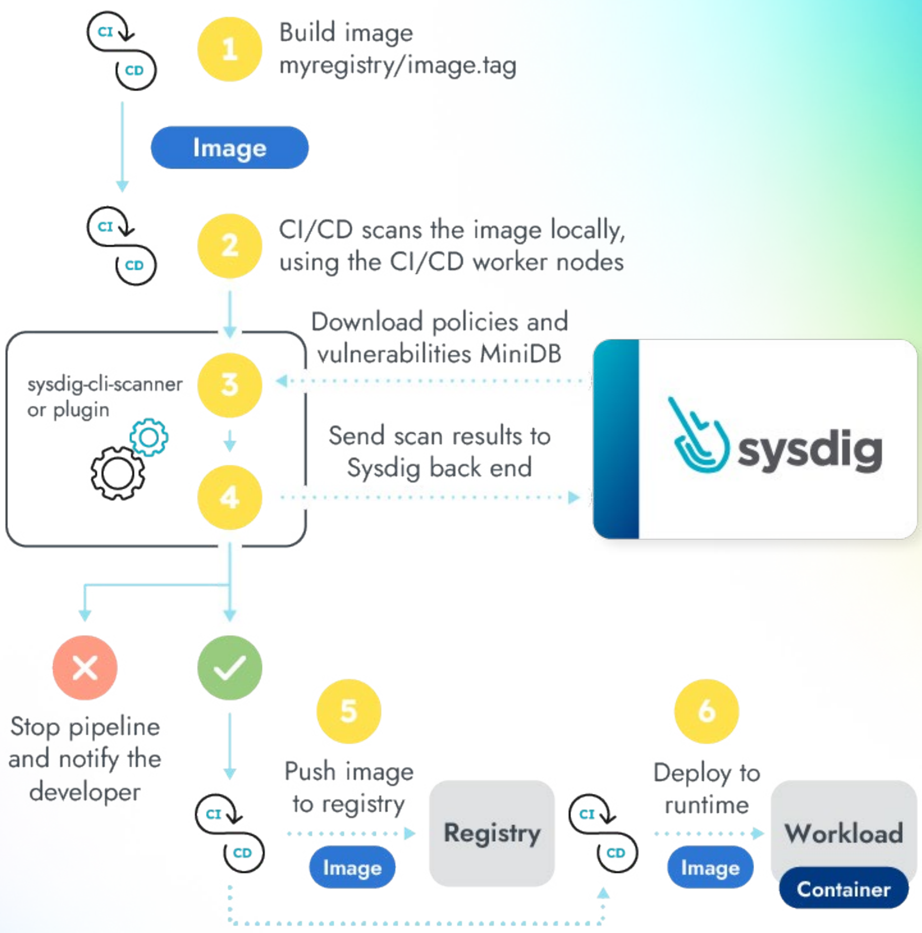
## Host Scanning

Hosts are a fundamental asset, as well as part of the potential attack surface. Vulnerabilities found in hosts that are supporting the execution of containerized workloads (such as OS vulnerabilities) can be even more pervasive and allow potential attackers to pivot or escalate. Sysdig's scanner supports the assessment of both traditional and container-optimized OSs.

Although host scanning itself is available in many forms, Sysdig's Vulnerability Management reduces operational friction by consolidating host, Kubernetes, workloads, containers, and every runtime asset using the same product in a single management plane for a seamless user experience.

## Container Image Scanning

Much like other forms of vulnerability scanning, container image scanning aims to identify flaws in software contained within a container image. Sysdig uses a software composition analysis approach to inspect the OS, libraries, packages, and other dependencies within each image, and to identify potential problems. The contents of the image are matched against credible sources of vulnerability data and then further filtered and prioritized with both static and runtime context.


CI
CD
1
Build image
myregistry/image.tag
Image
2
CI/CD scans the image locally, using the CI/CD worker nodes
sysdig-cli-scanner or plugin
3
Download policies and vulnerabilities MiniDB
4
Send scan results to Sysdig back end
sysdig
Stop pipeline and notify the developer
5
Push image to registry
Registry
6
Deploy to runtime
Workload
Container

## Vulnerability Data Sources and Feeds

Sysdig integrates highly reliable vulnerability feeds from both publicly available and proprietary sources in order to provide the most comprehensive and accurate scan results. The vulnerabilities are always indexed by a common vulnerabilities and exposures (CVE) ID for easier search and navigation. The data sources include:

- **OS vendors** (Red Hat, Canonical, etc.) provide the fastest and most complete information about flaws in their own products.
- **The National Vulnerability Database** is a U.S. government database, and the largest and most commonly used publicly available source of vulnerability information. It aggregates known vulnerabilities for all types of systems and assigns them a severity score based on the Common Vulnerability Scoring System (CVSS).
- **VulnDB** is a proprietary database offered by Flashpoint's Risk Based Security. VulnDB includes a broader scope of vulnerabilities, including some that have not yet been assigned a CVE ID. Additionally, VulnDB enriches this data with threat intelligence, including the vulnerability's exploitability status, links to proof-of-concept code, and more.

Sysdig frequently integrates additional sources to address specific use cases or improve the quality of the scan results overall.

## End-to-End Pipeline Scanning

The VM system is designed to enable scanning and vulnerability status tracking of container images throughout the software development life cycle (SDLC). The system includes the following capabilities, which become relevant at different stages of the SDLC and are best used in concert to provide maximum visibility and protection:

- **Local scans** on developer machines, performed by the sysdig-cli-scanner.
- **CI/CD integrations** through sysdig-cli-scanner and native plug-ins for Jenkins.
- **Registry scans** using the registry-scanner tool in the UI or through on-push triggers with native integration for Harbor, Artifactory, and other repositories.
- **Admission control** rules prevent the deployment of noncompliant images to the cluster, depending on image scan status.
- **Runtime scanning** continuously reevaluates running workloads for newly disclosed vulnerabilities.

> Sysdig integrates highly reliable vulnerability feeds from both publicly available and proprietary sources.

## Vulnerability Policy Engine

The Vulnerability Policy Engine allows for the definition of policies that evaluate a set of rules over a scan result and output a pass or fail verdict. The engine generates alerts when a policy fails. Because no environment is ever completely free of flaws, policies allow users to tune the threshold for what is acceptable within each scope, which also makes it easier to maintain and report on the vulnerability status of environments beholden to particular compliance requirements such as PCI DSS. There are two types of supported policies:

- **CI/CD policies** can be set to always apply, or users can explicitly match specific policies as part of each CI/CD pipeline.
- **Runtime policies,** including "host" scan policies, have a runtime scope like "entire infrastructure." Users can also make them more narrowly specific using deployment labels and environment metadata such as "host.name = foo" or "kubernetes.cluster = my-cluster and kubernetes.namespace in (ns1, ns2)."

Policies consist of a collection of rules. Many default rules ship out of the box, and it is possible to create custom rules to suit unique use cases. The currently supported rule types are:

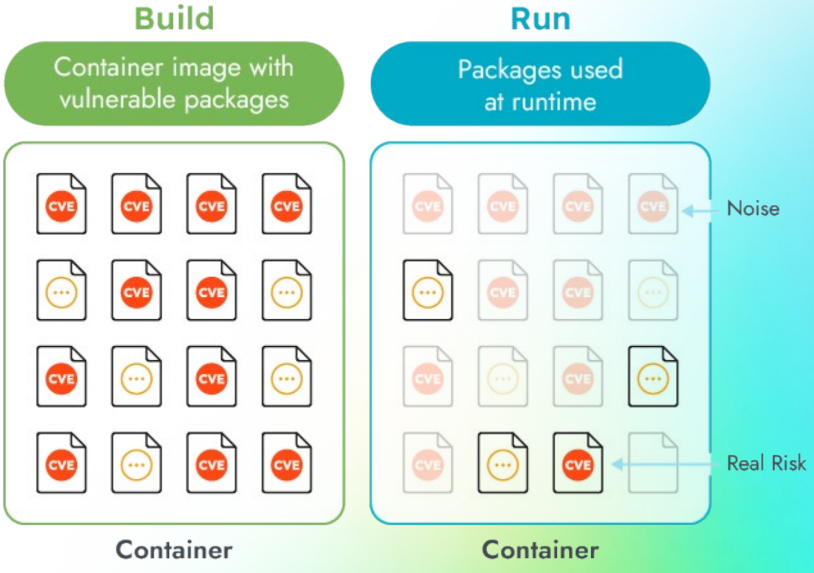
- **Vulnerability attribute rules**, which allow for triggers other than the standard severity score, fix available, publish and fix date, and so on. These rules enable users to also check the attack vector, whether an exploit is available, and much more.
- **Image configuration rules,** which allow checking the image's "effective user," the definition of environment variables with a secret pattern, and other configuration elements.

The engine evaluates policies on the client side when scanning the image. When integrated into the CI/CD pipeline, the engine evaluates policies anytime the inline-scanner is executed. For runtime workloads, there is a periodic rescanning of all running images, which will reevaluate the relevant policies.

## Risk Spotlight

In addition to the standard metric of vulnerability severity and context from VulnDB, such as exploit and patch availability, Sysdig offers unique runtime information about each vulnerability. Runtime profiling helps consider not only the theoretical impact of the flaw, but the actual risk that its presence may pose to the organization based on the application's real behavior.

Risk Spotlight leverages system call data to detect which packages are executed and flags them as "in use," enabling the prioritization and filtering



of vulnerabilities on packages that are actually executed in existing workloads. Highlighting "in use" packages can reduce the vulnerabilities that developers need to address immediately by up to 95%.

Ultimately, Sysdig's full set of prioritization features allows users to narrow down the list of the most dangerous and impactful vulnerabilities, and address those first.

## Reporting

Sysdig reports all vulnerabilities discovered on all assessed assets during a scan, and automatically sorts results to display the most critical vulnerabilities at the top of the list. Filters are available for further refinement by scope, exploit status, and other attributes. Every finding includes CVSS, exploit status, fixed version if available, and Risk Spotlight information.

Reports are scheduled, with configurable report attributes and filters along with the report schedule (for example, daily at 1 a.m.). When the time arrives, the system queues the report. A Generate Now option can also force a report into the queue without waiting for a scheduled time. After completion, the system sends a notification to a notification channel with a download link for the report. Reports are also obtainable via API.

The reports are in S3 storage, so the download link is a redirect, with a set of headers that temporarily allow downloading the report from the S3 bucket. Additionally, the API allows querying the list of executed reports, downloading an older report, and much more.

> Sysdig reports all vulnerabilities discovered on all assessed assets during a scan, and automatically sorts results to display the most critical vulnerabilities at the top of the list.

## Exceptions

Exceptions enable the temporary exclusion of vulnerabilities from generating alerts. This is a form of risk acceptance, for snoozing issues to address at a later date, or linking to a ticketing system for tracking.

04

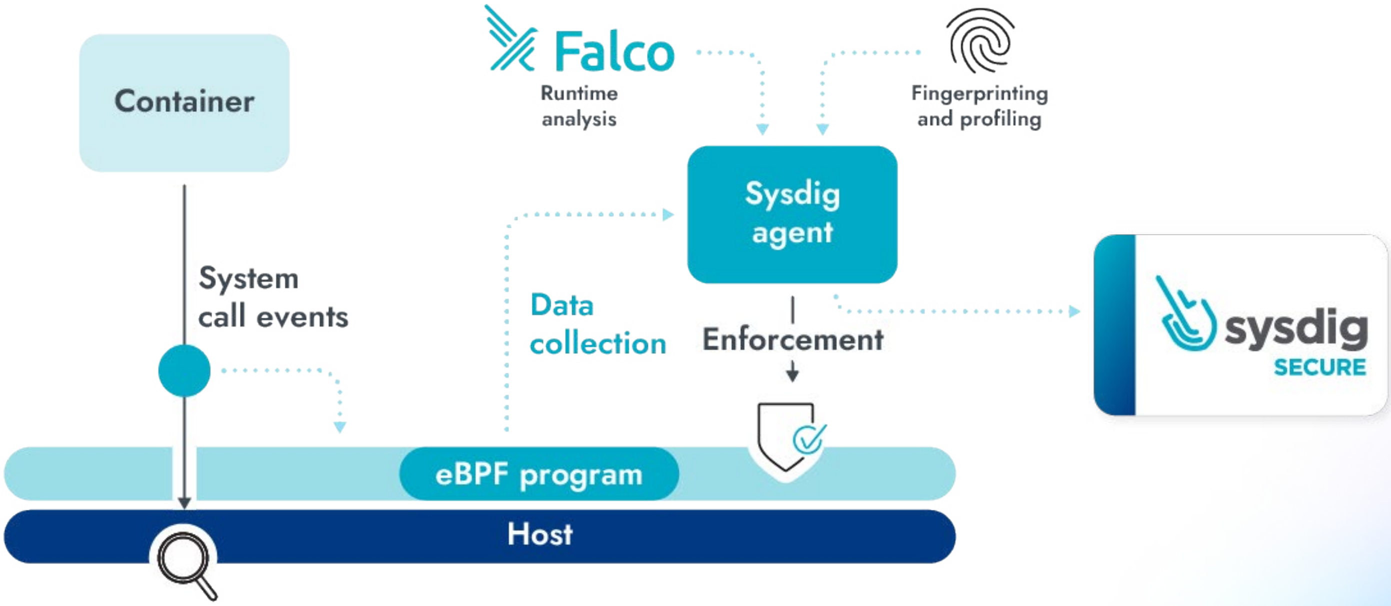
# Runtime Workload Protection

Runtime protection includes controls that manage a workload's security as it is running in production. Sysdig includes both preventive and detective controls for cloud hosts and workloads, including containerized applications.

Sysdig offers a range of capabilities to protect cloud and container workloads during and well beyond their initial deployment. In addition to monitoring the vulnerability and configuration state of both the infrastructure and the applications, Sysdig offers agent-based and agentless instrumentation for preventive and detective use cases. Sysdig can also detect threats against serverless functions. The tool is designed to accelerate incident response by providing a detailed record of activity enriched with metadata, enabling security operations teams to surface what happened, where in the cloud infrastructure, and who is responsible for the affected services.

## The Sysdig Agent

The Sysdig agent is the primary method of data collection for both infrastructure and workloads, including physical or virtual machines, IaaS instances, containers, and Kubernetes clusters. The agent instruments every host, either as a Linux service, container, or Kubernetes DaemonSet. It attaches a kernel module or eBPF probe to capture system calls.



Applications interact with the kernel and with other applications through system calls. Getting visibility into all system calls that traverse the kernel allows for the association of security events with system call activity including user behavior, file, process, and network activity. The Sysdig agent natively integrates into the container runtime API and Kubernetes API, enabling metadata collection and enrichment of generated events.

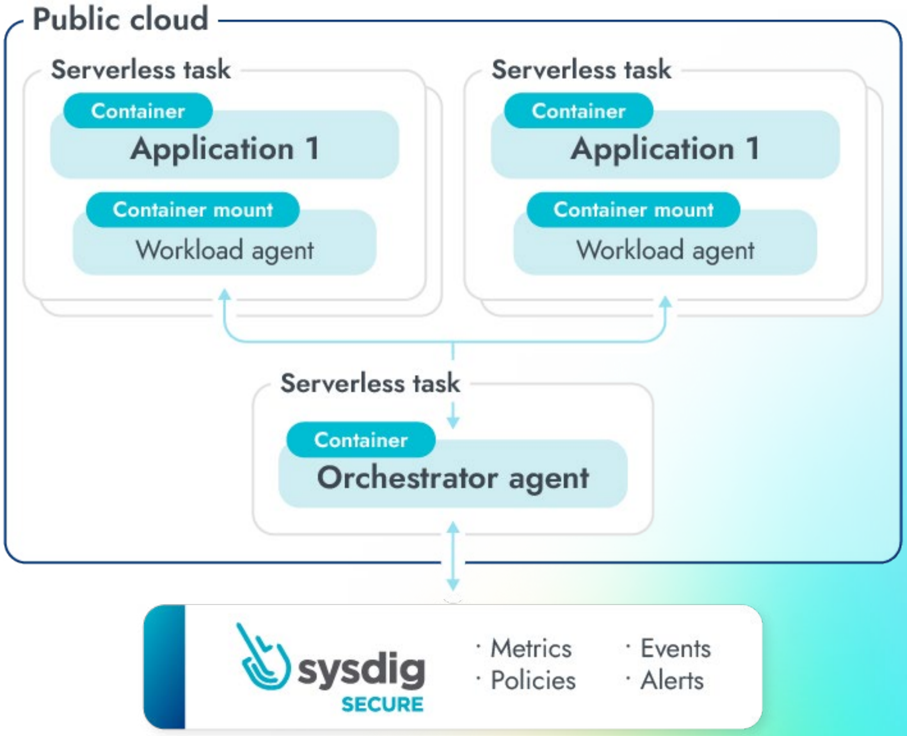
Having the Sysdig agent installed at the host level avoids the overhead associated with sidecar deployments that otherwise would need to be installed for every pod or container, severely impeding scalability.

Additionally, users can streamline events to a Security Content Automation Protocol file, and record and store those captures that contain data about what occurred before and after an event in a memory ring buffer for later forensic analysis (see the Threat Detection, Incident Response, and Forensic Analysis section).

## Serverless Agents

To enable fully cloud-native coverage, it is possible to deploy Sysdig with components that do not require direct host instrumentation:

- **Workload agents** are currently available for Amazon ECS Fargate. They are installed on each application's task, either by being baked into the application container or attaching a sidecar as the main process, with the application as the child process. Ptrace instrumentation captures system calls that a lightweight implementation of Falco then evaluates.
- **Orchestrator agents** are an optional but recommended component that works as a proxy between the workload agents and the Sysdig collector. This allows the application tasks to remain on a private network and communicate with the back end only indirectly through the orchestrator agent.



## Metadata Enrichment

One of the benefits of deploying cloud-native workloads is their vast scalability, but this creates a key challenge: how to manage and secure huge environments that contain a multitude of diverse and often ephemeral workloads. Sysdig is designed to offer all information about environments in a fully contextualized way, enabling users to make decisions about the appropriate response actions without any extra steps.

For example, any event generated by Sysdig includes extensive information about affected assets, including the container name, Kubernetes cluster to which it belongs, pod, namespace, and service/deployment. Most of the Kubernetes and container enrichment happens on the sysdig-agent before being sent to the sysdig-backend. Further enrichment is possible if there is a cloud account connected, to correlate things like AWS accounts, GCP projects, cloud resource types, security groups, regions, and so on. The events and their context are visible in the Insights dashboards according to their metadata hierarchy, frequency of occurrence, and severity.

> Sysdig is designed to offer all information about environments in a fully contextualized way

## Runtime Policy Engine

The Sysdig Runtime Policy Engine enables the creation of rules, alerts, response actions, exceptions, and a variety of other logic around the telemetry gathered by the agent and other data sources. Sysdig policies allow for preventive, detective, or reactive configurations.

### Rules

Rules based on the Falco language are added to a policy and pushed to an agent that evaluates events based on the source of the Falco rules (syscall, k8s_audit, aws_cloudtrail, and others). Rules contain three key elements: conditions, output, and exceptions.

- **Conditions** evaluate each event based on the source type. For example, system calls can evaluate things like the process name, user name, or arguments.
- **Output** may contain variables in order to give users the information they need to investigate, such as the process name, user name, or arguments.
- **Exceptions** are used to allow list an attribute or a set of attributes. For example, users can make a granular exception for a process name, or make more fine-grained exceptions like process name + user name + image name.

### Policies

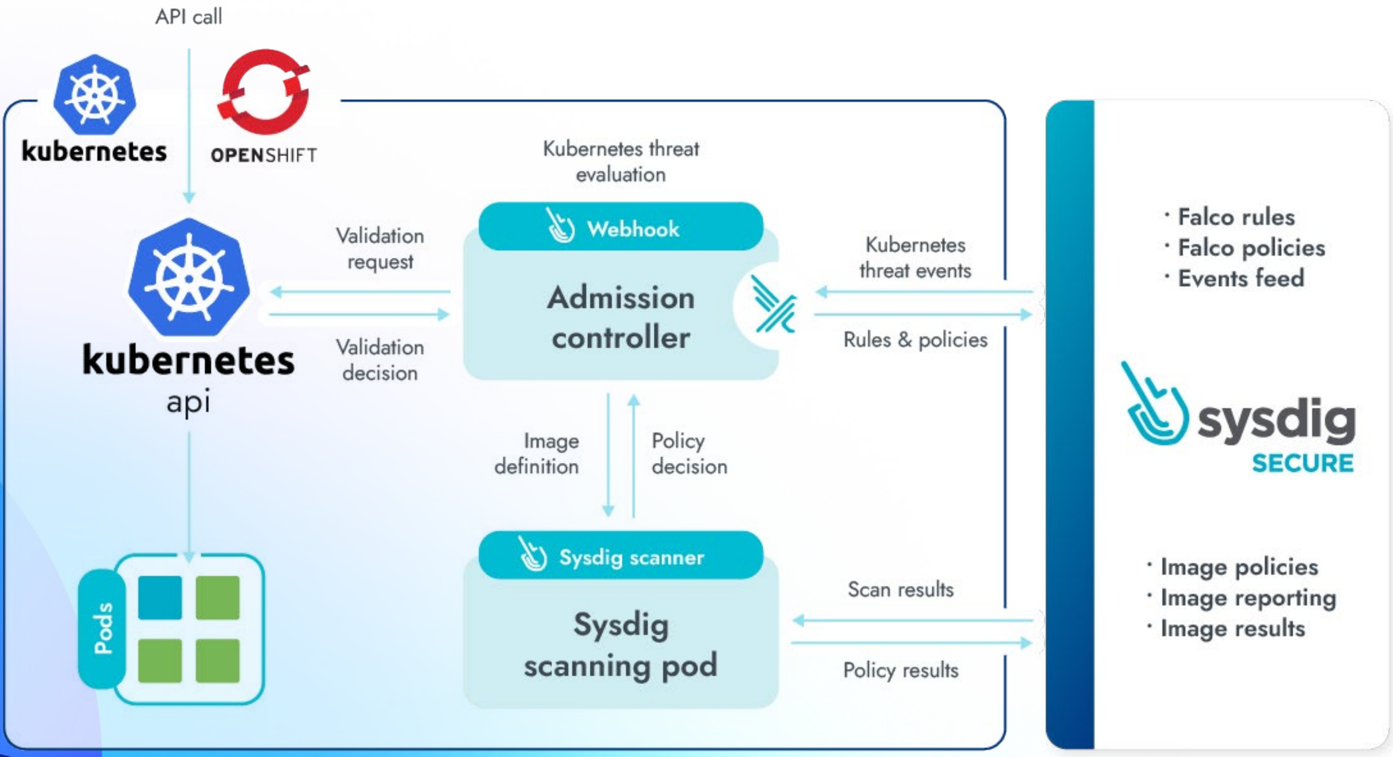
Policies group one or more rules to a scope with associated actions and notifications. The policy type dictates what scopes or actions to take based on an event. The available policy types are workload (syscall), Kubernetes (k8s_audit), cloud (aws_cloudtrail, gcp_auditlog, etc.), drift, and machine learning. All rules must have the same source, such as syscall, k8s_audit, and so on.

- **Scope** defines where to apply the policy. For example, a policy can apply to the entire infrastructure or a single Kubernetes cluster. Users can give workload policies nearly any tag or label, scope Kubernetes policies to cluster or namespace, and scope cloud policies to account numbers or regions.
- **Actions** are possible responses to an event. All policies can generate an action of sending the event to a notification channel. Container actions can pause, stop, or kill a container. Only workload policy types have capture actions, with the ability to collect system calls before an event occurred and response actions to kill a misbehaving container.
- **Severity** ratings of high, medium, low, or info are assigned to each policy out of the box by Sysdig's Threat Research team, based on the security risk associated with the event. Custom policies allow users to select the relevant severity level.
- **Notification channels** are configurable for all policy types. Users can have notifications sent to any defined and configured channels.

Some policies enable additional customization, such as the confidence level on machine learning detections. Sysdig ships with dozens of managed policies pre-built, which our Threat Research and Engineering teams frequently update. These include rules for detecting the most common cloud and container security bad practices and potentially malicious activity. Users can create their own custom rules and policies to suit specific use cases.

## Kubernetes Admission Controller

A Kubernetes admission controller is a mechanism for imposing a set of rules that govern the allowable requests to the API server. The controller can prevent certain workloads from running or send alerts for particular types of requests.



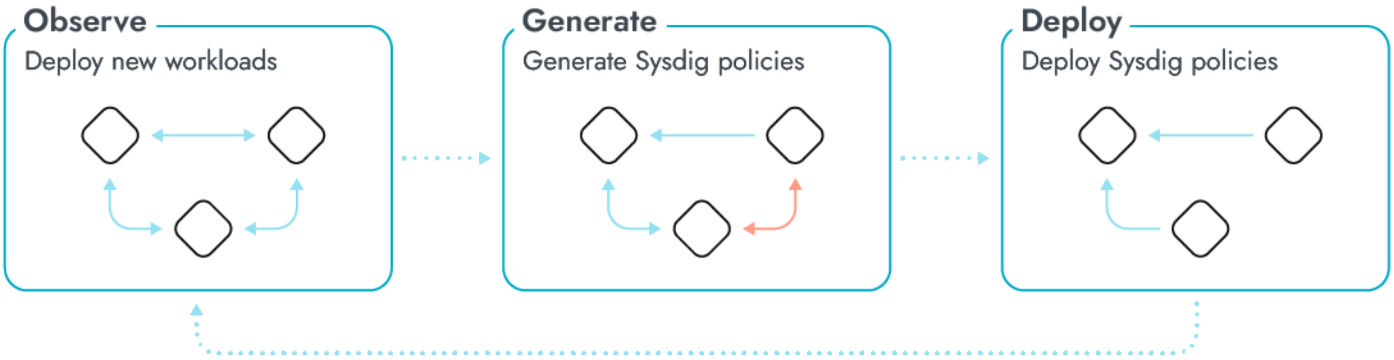
Sysdig supports these admission control features:

- **Kubernetes Audit Logging** inspects every Kubernetes API call, compares it against a Kubernetes runtime policy (k8s_audit), and sends alerts on violations.
- **Image Scanning** allows or denies a container image to deploy to the Kubernetes cluster based on its vulnerability status, including scan age, pass/fail scan policy, or whether the image was scanned at all.
- **Configuration Assessment** checks the workload's configuration against the relevant CSPM/KSPM policy before allowing its deployment to the Kubernetes cluster, blocking or sending an alert about the configuration if it is not compliant.

Admission controllers are very powerful, but should be used with some caution, as they can prevent workloads from running and even break a cluster if blocking critical or system workloads.

## Kubernetes Network Policy Advisor

Sysdig will report any observed traffic between pods and use this data to record a workload's normal network activity. Sysdig will provide recommendations for more restrictive network policies in accordance with zero-trust architecture best practices.



Admission controllers are very powerful, but should be used with some caution, as they can prevent workloads from running and even break a cluster if blocking critical or system workloads.

05

# Threat Detection, Incident Response, and Forensic Analysis

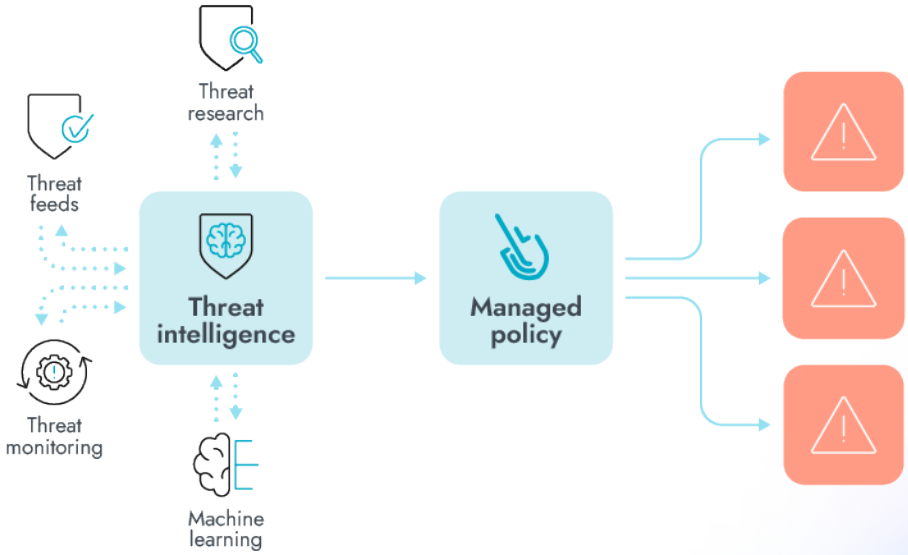
Completing the source-to-run security suite is Sysdig's open source-based threat detection engine built on top of Falco, the de facto standard for cloud and container threat detection. Sysdig offers an extensive and frequently updated out-of-the-box detection content library. Complementing detection capabilities with incident response and forensic analysis functionality enables teams to quickly triage issues and fully investigate known or suspected security incidents.

## Threat Detection

Sysdig enables detection capabilities with a flexible policy engine, which generates alerts or response actions based on telemetry gathered from multiple data sources, including system calls and logs. While the Runtime Workload Protection section describes its design, the following additional elements of Sysdig Secure form a complete cloud-native threat detection solution.

### Data Sources

The Sysdig threat detection system ingests data from multiple data sources to enrich events and trigger alerts. Supported data sources include system calls, orchestrator and container audit logs, and cloud logs.



### Detection Content

Detection content includes rule- and machine learning-based methods to identify and send alerts about suspicious behaviors in the environment. Sysdig includes an extensive library of detection content out of the box and frequently appends and updates this content. Additionally, Sysdig customers have access to the rules provided by the Falco open source community. Users can create custom rules to meet their unique needs using the Falco language or Terraform.

Threat intelligence informs special high-fidelity rules based on indicators of compromise, such as command and control Internet Protocol addresses, that can alert users to malicious activity in their environment. Sysdig provides rules based on open source, premium third-party, and proprietary Sysdig-developed threat intelligence. Rules update multiple times per day, with new content created in response to global security events like zero-day disclosure.

## Tuner

To improve the quality of the event feed, Sysdig offers two tuning mechanisms:

- **Automatic tuning** identifies the noisiest events and creates targeted exceptions to the rules to reduce unnecessary alerts. This method is very effective at excluding known-benign patterns. Automatic tuning is not enabled on high-fidelity rules, which are low-false-positive rules that Sysdig deems highly important.
- **Manual tuning** allows users to create exceptions to rules and tailor them to the unique aspects of their environment. Sysdig offers tuning suggestions in the Insights dashboard, which users can elect to enable based on extensive testing and threat research. Additionally, the Runtime Policy Tuning screen includes an editor to fully customize exceptions.

Automatic tuning is enabled by default for most rules, but many environments will benefit from some degree of manual tuning to ensure that the tool is fully tailored to unique organizational needs.

## Drift Control

Sysdig can detect the addition or modification of a binary after the container started. Drift Control allows users to configure alerting or blocking actions to prevent such a process from running. Users can add their own allowed or deny lists in addition to automatically detected binaries. Because most containerized workloads are immutable by design, container drift is often indicative of malicious tampering, human error, or noncompliance with best practices.

> Sysdig provides rules based on open source, premium third-party, and proprietary Sysdig-developed threat intelligence. Rules update multiple times per day, with new content created in response to global security events like zero-day disclosure.

## Machine Learning and Advanced Analytics

Sysdig uses machine learning and other advanced analytics approaches to supplement rule-based detection. Machine learning can offer unique advantages for use cases that involve large reference data sets and highly repetitive patterns not easily recognizable by human analysts. Machine learning is also effective for advanced baselining and anomaly detection. Sysdig ships with these features:

- **Image profiling** monitors workloads over time and baselines the observed system calls, network traffic, and other attributes. This enables alerts based on deviations from expected behavior, particularly for immutable workloads.
- **Cryptominer detection** is based on advanced machine learning algorithms that are trained to extract and send alerts about features not easily detectable by rule-based approaches.

Hybrid approaches are the most effective in providing comprehensive threat detection coverage. Sysdig's machine learning detection system is designed to complement the rule-based policy engine for optimal results.

# Incident Response and Forensics

Sysdig is designed to aid in preventive and detective processes, as well as the investigative actions required when a security incident occurs. As such, the tool provides multiple features to collect and explore data within the interface, in addition to the option of exporting data to third-party systems.

## Response Actions

Sysdig enables multiple response actions upon the detection of suspicious activity. Users can configure policies to send alerts on such activity or kill the affected workload. Additionally, a capture can record activity before and after the event was triggered for further investigation. It's also possible to connect directly to the affected workload from the UI to troubleshoot the issue in real time.

- **Captures** are files that contain system call data, which users can configure to retain data for any length of time before and after the event. Captures use a ring buffer to constantly collect runtime information on workloads (either hosts or containers), and allow users to inspect processes, related threads, involved files, and network flows. Captures can automatically trigger in case of a detection event, as part of a policy configuration, or users can perform captures proactively.
- **Rapid Response** allows quick initiation of a shell into a special container, which can mount volumes or service accounts on the host to enable access to the Docker API, Kubernetes API, or the entire host file system. This functionality is installed as a separate component, making it possible to enable it only for certain systems. Additionally, users can configure Rapid Response to be accessible to dedicated groups only, in order to prevent mistakes and tampering.

## Investigation Flow

There are multiple avenues for investigating events within Sysdig Secure, which provide different types of information or enable different types of workflows.

- **Events Feed** lists all security events in chronological order, grouped by policy, and provides filtering options to navigate through them. The Events Feed is also visible through the Insights dashboard, which displays all of the context and detail associated with an event.
- **Activity Audit** provides a listing of all commands from interactive sessions, file operations, and network operations details for listed commands. It also gives a brief vision on interactive processes that happened before, during, or after detection.
- **Events Forwarder** enables the exportation of event data to a third-party system, such as a security and event management system. The data sources that can be forwarded are activity audit, benchmark events, secure events compliance, host vulnerabilities, runtime policy events, and Sysdig Platform audits. The currently supported integrations are:
    - Amazon Simple Queue Service
    - Elasticsearch
    - GCP Chronicle Security
    - Google Cloud Pub/Sub
    - HTTP Event Collector (Splunk)
    - IBM Multicloud Manager
    - IBM QRadar
    - Apache Kafka (only for on-premises installations)
    - Microsoft Azure Sentinel
    - Syslog
    - Webhooks
- **Capture Inspect** displays the data collected by the Capture function in the Sysdig UI and allows a variety of filtering options, including filtering by attributes of containers, file systems, network connections, processes and system calls.

**Sysdig is designed to aid in preventive and detective processes, as well as the investigative actions required when a security incident occurs**

# Platform Features

## Authentication

Sysdig offers manual user configuration or single-sign-on (SSO) integration. SSO gives users the ability to use a single, centralized set of login credentials for those accessing Sysdig applications.

Sysdig offers these SSO options:

- Google OAuth (SaaS, on-premises)
- Security Assertion Markup Language (SaaS, on-premises)
- OpenID Connect (SaaS, on-premises)
- Lightweight Directory Access Protocol (on-premises only)

## Teams

Organizations can use Sysdig Teams to give teams access to only those applications for which they are responsible, while platform teams maintain a federated view of their global clusters and workloads. In addition to restricting visibility, the platform offers role-based access control for fine control of individual users' access to the product.

## Roles

Sydig allows for the grouping of fine-grained permissions into rules applicable to users within a scope of teams. Most permissions follow the standard read, edit, exec pattern, although there are some exceptions.

## Subscription

Subscription details provide an instant overview of current subscription status, usage, and licensing information, including reserved agents, agent utilization, license type, and more.

## Login Message

Login Message offers the ability to define a "terms and conditions" type of message for users to accept before accessing the system. The message supports markdown for formatting.

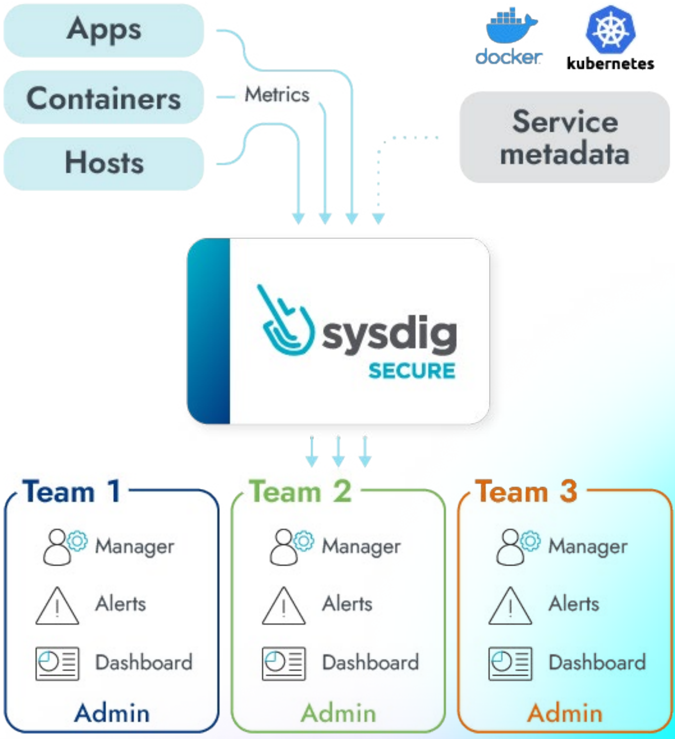
## Sysdig Platform Audit

Sysdig provides a set of APIs for auditing and reporting on the use of the Sysdig platform itself.

The audit includes the following request methods against the Sysdig system:

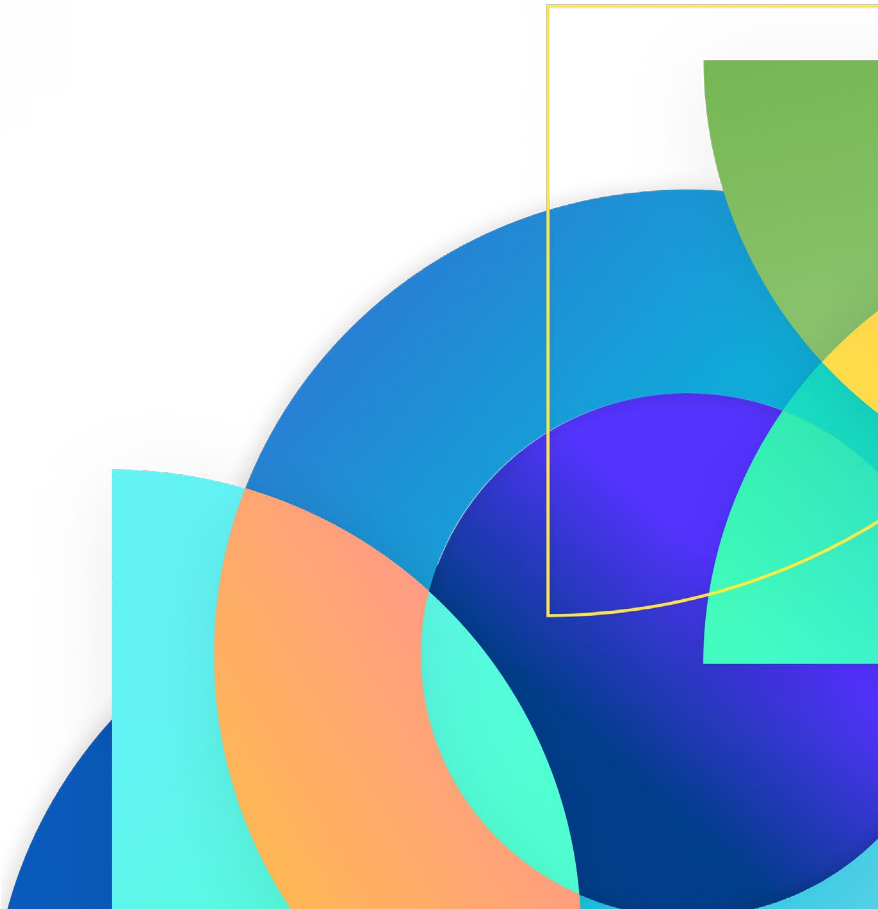
- Put
- Post
- Delete
- Patch
- Get

The data retention for system audit information is 90 days.



# Additional Resources

For more detailed information about Sysdig's portfolio of products, see our documentation page at docs.sysdig.com, or access our customer support function at sysdig.com/support.



sysdig

Guide-015 Rev. A 12/22.