WIZ | Google Cloud

Accelerate your cloud journey securely with Wiz and GCP

A solution for Google Cloud Container Security both security and developers can love

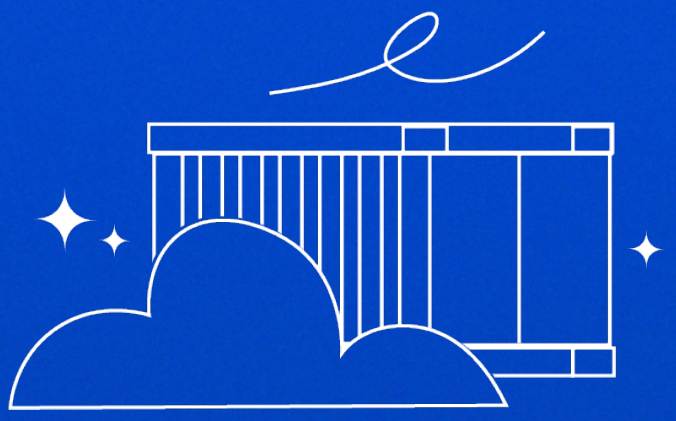


Table of Contents:

Achieve cloud security at scale through agentless, full-stack, and contextual risk assessment

The Log4Shell vulnerability posed an urgent security crisis for countless organizations, and especially those using Google Cloud containers. Discovered in December 2021, the flaw in the Log4j Java logging framework made it possible for hackers to take control of countless enterprise applications. Fast remediation was critical, but even security teams with container security solutions in place couldn't always determine which of their containers were running with the framework. The situation underscored the need for a new approach to Google Cloud container security—one allowing better visibility into the dynamic environment, more context into the risks present, and a more effective way to mobilize developers quickly to close security gaps.

This white paper discusses the Wiz solution for Google Cloud container security. Providing complete agentless visibility for containers in the Google Cloud environment, as well as deep risk assessment for the network, the identities that can access it, and the cloud it runs on, Wiz helps organizations understand, prioritize, and communicate security vulnerabilities in Google Cloud containers.

With coverage across any container and orchestrator, including GKE in Google Cloud, Cloud Run Serverless container self-hosted Kubernetes on Compute Engine instances, and containers on standalone virtual machines (VMs), the solution offers protection against Log4Shell-like threats no matter where the container runs in Google Cloud. By giving developers the tools to fix issues earlier, faster, and more easily, Wiz enables a shift-left approach to security and compliance for container images across their lifecycle.

Why containers are transforming enterprise technology

Containers are transforming the modern technology environment. Encapsulating both an application and all the elements it needs to run—system libraries, system settings, and other dependencies—containers offer the benefits of a virtual machine (VM) in an even more compact and portable form. They're repeatable and standardized, running the same way wherever they're hosted across clouds and operating systems, making them simple to move around and deploy wherever and whenever they're needed. As a fast, flexible way to deploy applications, APIs, and microservices, containers have quickly become a core element of Agile and continuous integration/continuous delivery (CI/CD) methodologies.

As organizations embrace containers, they turn to Kubernetes as the preferred tool to deploy, scale, and manage containerized applications across any type of infrastructure. According to the 2021 Cloud Native Computing Foundation (CNCF) annual survey, 96 percent of organizations are either using or evaluating Kubernetes. With 5.6 million developers already using Kubernetes, or nearly one-third of all backend engineers, it's clear that Kubernetes is here to stay. That makes Kubernetes security a top priority.

Security challenges with containers

Containers can help organizations achieve radical gains in the velocity and value of innovation, but they also bring new challenges for both security teams and dev teams. In a classic good news/bad news scenario, containers can be configured much more flexibly than a VM. They're able to do everything from mounting volumes and directories—to disabling security features. In a "container breakout" scenario, when container isolation mechanisms have been bypassed and additional privileges have been obtained on the host, the container can even run as root under the control of a hacker.

The powerful (and nerve-racking) flexibility of containers makes it essential for security teams to know exactly when and where they're running, and fully understand the risks they might be introducing. For development teams, however, leveraging the full value of containers depends on speed, agility, and self-service. The last thing they want is to get caught in slow-moving preventive security processes by security teams they don't trust to understand the technology. As a result, developer-security friction, a constant theme in today's world, can be especially acute when it comes to containers.

As security teams try to secure their organization's Kubernetes environments, they face several critical challenges.

Maintaining visibility

Dynamic, constantly changing Kubernetes environments are extremely hard to keep track of, with development teams spinning up new clusters and new workloads inside clusters on a daily basis. Security teams lack the necessary tools to understand what the environment looks like at any given moment, making an already difficult job even more challenging. This was a huge problem during the Log4Shell crisis.

Gaining context

Alert fatigue is real. With different tools for monitoring workloads, entitlements, compliance, vulnerabilities, and more, each generating its own alerts, security teams can lose important signals among the noise. With no context for the alerts, there's no way to prioritize a response, or to convince developers that it's important to do so.

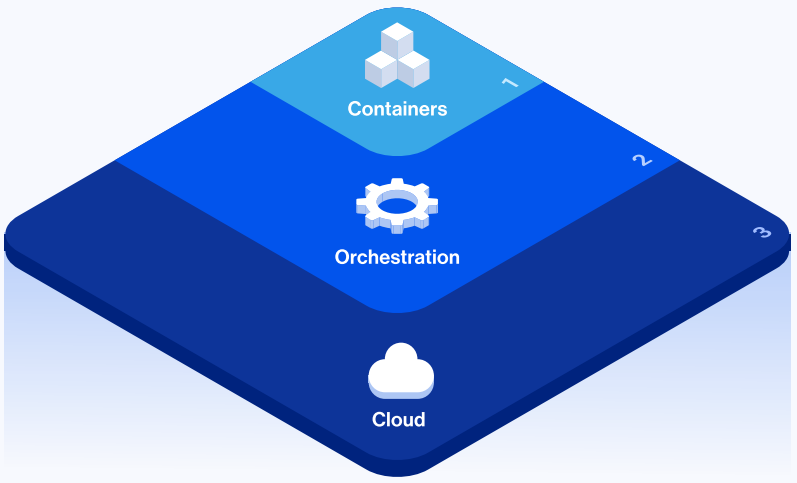
Understanding risk

In simple terms, container security encompasses three levels of risk.

- **Level one:** Securing the container itself. Security teams need to make sure that containers don't have vulnerabilities, and that they haven't been misconfigured to run on their host with permissions that are too powerful.

- **Level two:** Container orchestrators risk factors. Orchestrators, such as Kubernetes or GKE in Google Cloud, manage the network and identification layers used by containers. The wrong configuration can expose containers to the internet or grant excessive permissions in the cluster—or in the cloud itself.

- **Level three:** Cloud risks. In Google Cloud, most containers are orchestrated by Kubernetes or GKE. This means that a successful attacker can compromise an Compute Engine instance or another cloud service, and then laterally move to resources inside these clusters to use compute resources or access data.



To understand the risks associated with a container, it's not enough to focus solely on the container itself. You have to take into account the network, the entitlement, and cloud as well—making an already challenging task all the more daunting.

Organizational factors

As is so often the case, container security is undermined by a fundamental disconnect between security and development teams. Most security professionals are using siloed tools causing a fragmented view of their container and cloud security. As a result, they lose valuable time trying to find who is responsible for what in the environment. When they do manage to determine this, the tickets they raise lack the context needed to motivate developers to action.

This disjointed and inefficient process slows the development cycle rather than helping developers solve issues proactively. To ensure fast, effective container security at scale, security teams need to get developers on their side. That means finding an approach to container security that won't slow development to a crawl.

The need for a better container security solution

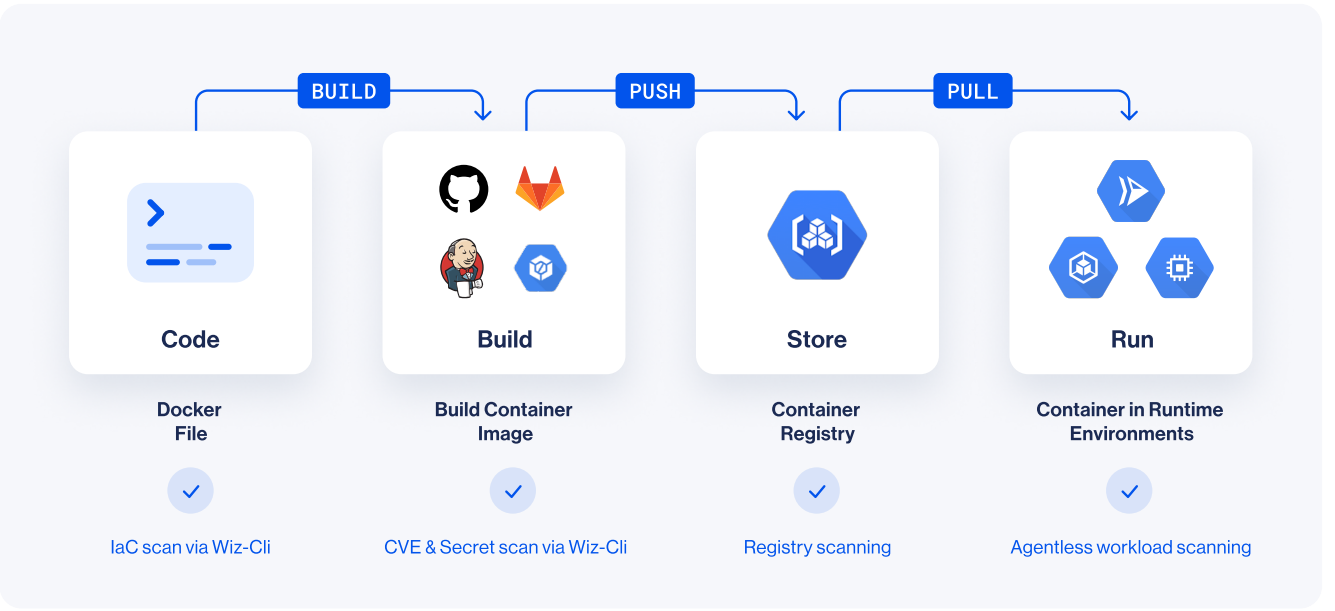
To date, security teams have been ill served by the available solutions to secure container environments. Existing container security solutions generally require the installation of an agent on the running resource controlled by the DevOps team. But both security teams and developers have concerns about agents, and understandably so. For security teams, agents require considerable operational overhead while leaving doubts about the quality of their coverage. For developers, agents can break things or change an application's behaviors. A high risk of blind spots, combined with the dynamic nature of the cloud environment, makes it impossible to achieve central visibility into all running containers.

Even with decent agent coverage in the environment, security would lack the context needed to understand risk. Existing siloed tools can't correlate the three types of risk—container, Orchestrators, and cloud—to provide security teams with the insight they need to prioritize issues. And to shift from reactive to proactive, and become an enabler rather than a blocker, security teams need tools that help them keep up with the rapid pace of development. To enable a shift-left approach to security, they need to embed risk reduction in the development pipeline.

What's needed is a single solution that assesses risk across the full container stack, provides context and insight, and gives developers the tools to fix issues earlier and faster.

How Wiz solves Google Cloud container security

Wiz gives security teams and developers a complete solution to manage container security across their lifecycle. Assessing and correlating risk across container images, identities, the Kubernetes network, and the cloud environment, Wiz enables comprehensive, end-to-end Kubernetes security posture management (KSPM) and compliance. As an agentless solution, Wiz provides complete visibility without compromising containers. Developer tools enable shift-left security for container images across their lifecycle.



Easy to deploy and use

The agentless Wiz container security solution connects to the Kubernetes environment with a simple API connector. Security teams can discover and scan containers, hosts, and clusters across cloud-managed and self-managed Kubernetes environments, including serverless containers such as Cloud Run as well as standalone containers running on VMs.

Risk-based prioritization

Before organizations can manage risk, they need to understand it. Wiz helps security teams determine where the greatest dangers lie by scanning the full container stack, mapping its service accounts and network configurations, and performing analysis into its cluster structure and context, network, identities, and secrets. Wiz also looks at the workloads themselves, including applications, containers, and VMs. All the information from these scans is collected and correlated within a security graph that helps security teams understand their connections. Instead of thousands of raw signals, security teams can quickly see the top risks that need to be mitigated to prevent a breach.

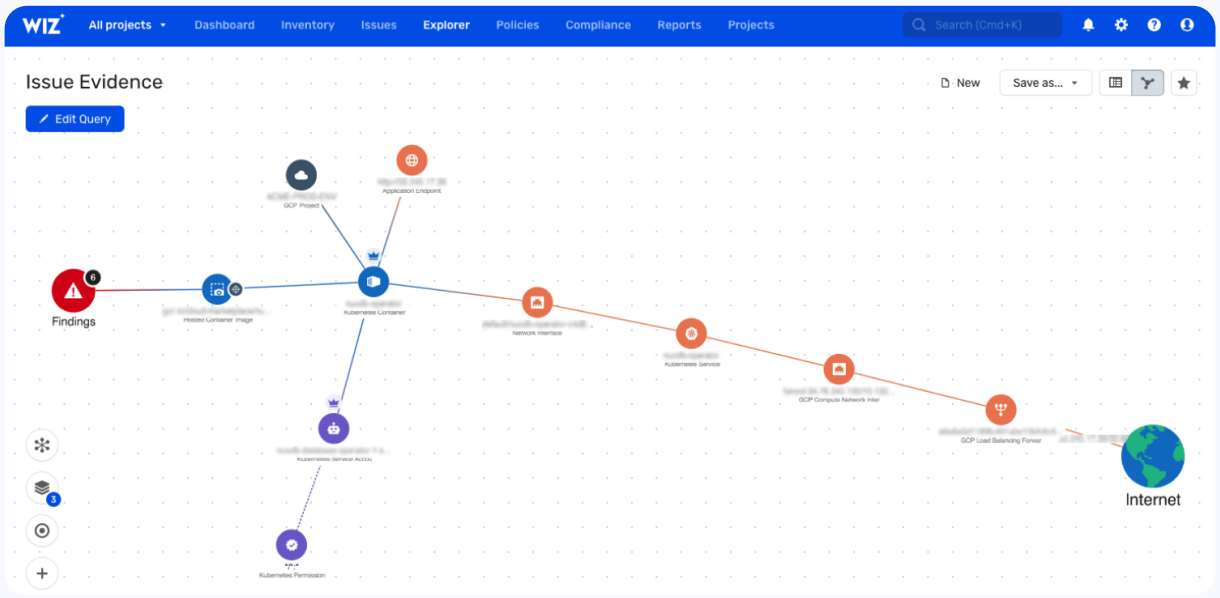
DevSecOps support

With an easy-to-use tool to secure container images and detect vulnerabilities and exposed secrets as part of their usual CI/CD pipeline, development teams can work autonomously to prevent security risks. Security teams can avoid slowing the development process while ensuring that uniform security policies are being maintained across the cloud and container development lifecycle, from the build stage to the registry to the runtime environment.

Why Wiz succeeds where traditional container security tools fall short

Working through an API rather than agents, Wiz provides visibility across the whole cloud environment and delivers full stack analysis in minutes without getting in the way of developers. The Wiz Security Graph offers crucial insight including:

- **A 3D visualization** with a single data layer across all the organization's clouds and components to understand what's happening in their environment
- **Insight beyond the containe**r to encompass an entire view of the cloud and workloads
- **Effective network exposure** showing effective paths to and from the container
- **Effective permissions** to understand the full scope of the risk associated with each container



Wiz Security-Graph visual of a publicly facing container running on Kubernetes with a High/ Critical network vulnerability with a known exploit and with high Kubernetes privileges

In this way, Wiz enables security teams to:

- **Identify and prioritize risk** k by scanning the entire environment, performing deep analysis, and providing actionable context so security teams can identify the most important issues, then push them out to developers with clear steps for mitigation. Wiz is uniquely able to surface toxic combinations which show how different risk factors might combine into an attack path to critical resources to build a single prioritized view of risk for an organization's environment. Insight beyond the container to encompass an entire view of the cloud and workloads.
- **Enable full-lifecycle security** by showing developers the security posture of their build both before and after it's stood up. Empowered to manage their own security posture, developers can take a shift-left approach to risk and play a valuable role in the shared security model.
- **Get the whole organization involved** d in container security. Full context into risks motivates developers to fix issues, while tooling such as an API console and command-line interface (CLI) helps them do so more quickly and easily. As a result, organizations can move faster with less risk.

The Wiz container security solution is part of a complete solution for full-stack security on Google Cloud, encompassing Security Posture Management (CSPM/ KSPM), Workload Protection (CWPP), Vulnerability Management, Infrastructure Entitlement Management (CIEM), CI/CD security (IaC, VM/container image, registry scanning), and Cloud Detection and Response (CDR).

Providing complete visibility into the entire Google Cloud environment across workloads, accounts, and environments, Wiz enables hundreds of organizations worldwide, including 20 percent of the Fortune 500, to rapidly identify and remove critical risks in their cloud environments. Its customers include:

To learn more about the Wiz solution for Google Cloud container security, visit our website and get a free demo.

About Wiz

Wiz secures everything organizations build and run in the cloud. Founded in 2020, Wiz is the fastest-growing software company in the world, scaling from $1M to $100M ARR in 18 months. Wiz enables hundreds of organizations worldwide, including 30 percent of the Fortune 500, to rapidly identify and remove critical risks in cloud environments. Its customers include Salesforce, Slack, Mars, BMW, Avery Dennison, Priceline, Cushman & Wakefield, DocuSign, Plaid, and Agoda, among others. Wiz is backed by Sequoia, Index Ventures, Insight Partners, Salesforce, Blackstone, Advent, Greenoaks and Aglaé. Visit https://www.wiz.io/ for more information.