CLOUDBOLT

CASE STUDY

How an enterprise software company moved from Kubernetes cost reports to automated rightsizing on 700+ clusters

A global enterprise software company used Kubecost to see what its Kubernetes workloads cost, but engineers rarely changed anything. It connected cloud cost to customers and margin with CloudBolt, handed rightsizing to StormForge, and put cost data in front of engineers inside the tools they already use. Automated rightsizing now runs on more than 700 clusters, and some engineering teams have cut their spend by more than half.

At-a-glance

CUSTOMER

Global enterprise software company that delivers its platform as a service

SCALE

More than $100M in annual cloud spend, more than 1,000 Kubernetes clusters across regions around the world

CLOUD MIX

Mostly AWS, plus GCP, Azure and some on-premises infrastructure

PRODUCTS

CloudBolt cost allocation, StormForge Kubernetes rightsizing and MCP server

REPLACED

Kubecost

KEY RESULTS



Automated rightsizing on more than 700 clusters, with production next



Cost and margin traceable to every customer, environment, and microservice



Engineering teams using cost data in their agent workflows cut spend by more than 50%



GKE on GCP in the same view as AWS

Challenge

A global enterprise software company spends more than $100 million a year on cloud, much of it to run its platform for customers in heavily regulated industries. Its engineers could already see what their Kubernetes workloads cost. They rarely did anything about it. Teams set generous CPU and memory requests to stay safe, finance asked for lower bills without knowing which services drove them, and cost reviews tended to end in finger pointing.

The company's cloud FinOps team set out to change that. It connected cloud cost to customers and margin with CloudBolt, let StormForge make rightsizing changes automatically, and brought cost data to engineers where they already work.

Where the company started

The footprint is large and moves quickly. More than 1,000 Kubernetes clusters run across AWS, GCP and Azure in regions around the world, and the company is partway through a multiyear move to a Kubernetes architecture. Its platform is also highly configurable. Customers tailor it to their own use cases, so no single CPU and memory default fits. Most workloads were overprovisioned, and some were underprovisioned, which put performance at risk.

The company used Kubecost for Kubernetes cost allocation, expecting engineers to act once they could see where container spend went. Most didn't, partly because they didn't trust the numbers. Engineers disagreed with how Kubecost scored efficiency, and its figures included discounts engineering had no control over. When its self-hosted data pipeline fell behind, dashboards showed costs spiking overnight. Its cost breakdowns also weren't tied to actual usage, so the company couldn't get accurate unit costs by customer.

Even when teams knew a workload was wasteful, they didn't know what to set its requests to. Rightsizing changes went out by hand about once a year, with little data behind them and no clear owner, so savings initiatives stalled. Tagging wasn't consistent across accounts, which made tracing a cost to a team or customer manual work, and with regulated customers, every figure had to be auditable.

Connecting cost to customers and margin

The team started by building one data model that connects the cloud bill to the business. CloudBolt's allocated cost data flows into the company's data lake, where it joins telemetry, the CMDB and business records such as contracts, entitlements and cost centers through shared keys like account, customer and contract IDs. A dollar on the bill can now be traced to a customer, environment, product or single microservice. Because those costs are tied to usage telemetry, unit costs per customer reflect what each customer actually consumes.

That gave the company contribution margin at every layer of the business. It also showed where margin would go as customers grew. Some accounts looked very profitable only because customers were using part of what they had licensed, and the team can now model what happens as usage catches up.

Handing rightsizing to StormForge

Seeing the numbers hadn't changed behavior, so the company brought in StormForge to make the changes itself. StormForge adjusts CPU and memory requests automatically within safe limits, scaling workloads down where they're oversized and up where they're short on resources. It sets requests from each workload's actual usage and keeps adjusting them, so there's no once-a-year manual pass.

For a company whose customers are in heavily regulated industries, letting software change resource requests on its own was a high bar. The team cleared it one environment at a time, starting with non-production workloads. Automated rightsizing now runs on more than 700 clusters, and its recommendations feed the capacity forecasts behind the company's savings plan and reserved instance purchases.

> "We have more faith in what **we're provisioning by default**, knowing that we can lean on those automated rightsizing events to protect us as we grow."
>
> **FinOps leader**
> Global Enterprise Software Company

Putting cost data where engineers work

The company's engineering teams increasingly build with AI agents, so the FinOps team brought cost data to them. Engineers query cost through the StormForge MCP server and an internal agent built on the company's data lake, and cost anomalies show up as work items on the owning team's board. Teams that often didn't know which cloud resources they had deployed can now see them, along with what they cost, without leaving their workflow.

The same data covers AI usage. Tokens, models and features map to customers and contract entitlements, with cost per million tokens and cost per interaction for each product.

Results so far

Cut spend by 50%

Engineering teams that started working with cost data in their agent workflows cut their spend by more than 50% within about a month. One core product team used the AI cost data to find that its most expensive AI features were the ones customers valued least. It changed the prompts and models behind them and cut those features' costs by more than half.

Unit economics

Leadership now works from unit economics. Cost per customer and cost per environment inform capacity planning, commitment purchases, product priorities and build-versus-buy decisions, and VPs from cloud, product and finance review the largest opportunities together each month. StormForge also brought GKE workloads on GCP into the same view as AWS.

FinOps ownership

For the FinOps team, the bigger result is ownership. Service teams see what they provision, what they use and what it costs, and cost conversations start from numbers everyone agrees on.

What's next

Automated CPU rightsizing moves to production workloads in the coming quarter, rolled out slowly. The company is also adopting StormForge's upcoming efficiency metrics to separate savings from automated rightsizing from savings from manual work. Along with provisioning customer environments at what they use and growing them toward their entitlement, those steps are part of the roadmap to cut production cloud costs by more than 10% over the next year.

If your teams can see their Kubernetes costs but aren't acting on them, talk to CloudBolt about how this team got there.

Contact us

CLOUDBOLT

© 2026 CloudBolt Software. All rights reserved.