

AI-Driven Kubernetes

Overcoming the FinOps Challenge
for Kubernetes



Realizing the Promise of Kubernetes: Efficiently Developing and Deploying Apps Anywhere

When it comes to app development, engineers have their eye on the prize: delivering software fast and reliably with innovative capabilities that drive business value (and enjoying the process).

The widespread adoption of Kubernetes, and the portability and flexibility of containers, has fundamentally changed – and accelerated – how developers build and deploy applications in multi-cloud environments.

They're not generally thinking about defining resource requests and limits for apps delivered via Kubernetes, and they are rarely empowered to understand the true costs of running applications. After all, without the ability to know the ideal configuration to minimize costs, how are they going to figure out the right answer?

**Quality (64%) and development velocity (53%)
are the top two reasons companies decide to
implement Kubernetes.**

SOURCE: [STUDY: CIRCONUS, SURVEY: THE COMPLEXITIES OF KUBERNETES, 2020](#)

But, it's not really just about managing and controlling costs when it comes to deploying apps on Kubernetes. It's about realizing the promise of cloud-native architectures and Kubernetes – fast time-to-market, superior user experience, effortless scalability, improved cost efficiency, and always-on availability.

To realize this promise, engineers need to be empowered and enabled to make smart resource decisions that minimize the cost of running applications – and the time and effort spent making those decisions – while ensuring business goals are met.

Decisions Need to Move as Fast as Modern App Development

In a containerized world, resource procurement decisions are decentralized. The absence of a centralized and formal capital expenditure approval process enables the engineers who build, deploy and own software to make decisions about the resources needed to run an application.

At the same time, the move away from hardware-constrained data centers to instances that can be turned on and off as rapidly as needs demand has shortened decision making timelines.

48%
of cloud spend is
WASTED
on average



SOURCE: [STORMFORGE, 2021 CLOUD WASTE SURVEY](#)

With the shift to cloud, capacity management processes also become less structured. The agility and speed enabled by containers and microservices give developers the ability to work more quickly and to make decisions that change application resource needs. Often, engineers are making resource decisions on the fly, in real-time, as needs and traffic increase or decrease.

Without the ability to take ownership of cloud resource usage, and make informed resource decisions quickly, organizations spend millions of dollars on wasted cloud resources, suffer from business-impacting performance issues, and lose thousands of hours of engineering productivity every year.

Aligning Engineering and Finance to Support Innovation

In today's innovation-driven environment, engineering and finance aren't always aligned.

While engineers understand the need to use resources – and time – efficiently, they also want to deliver software quickly and reliably with differentiated capabilities that create business value.

Finance, on the other hand, is focused on accurately forecasting and predicting spend, charging back and allocating costs to appropriate departments, while also controlling and (preferably) reducing costs. In short: it's all about budget and risk.

I just want to be sure my app works well and delivers a good user experience.

I don't have time to focus on the financial side of things...



ENGINEERING TEAM

Where is our cloud spend going? Why is it so high? Does it need to be that high?

We need to cut budgets by 30%...



FINANCE TEAM

Resource Management: Kubernetes Increases Complexity

With Kubernetes, engineers have even more decisions to make when deploying applications, and each of those decisions can affect app performance, reliability, and, of course, cost.

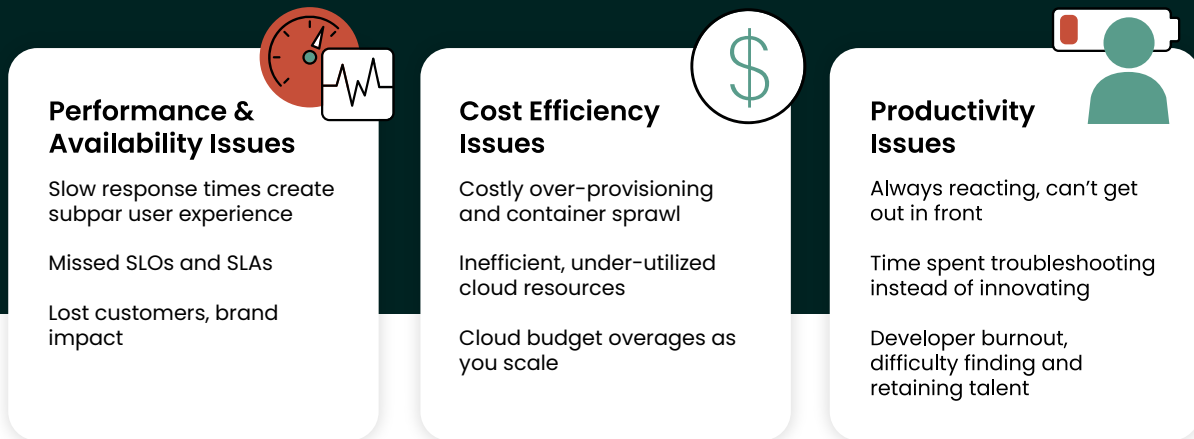
Every change has an impact, and the sheer number of settings that need to be configured when deploying a containerized app can result in an essentially infinite combination of configurations.

While a number of tools exist to help engineers address performance, such as load testing and observability platforms, they're all about identifying or troubleshooting problems. But, what do you do when you discover a problem? What can you do to address that problem? And, how might those actions impact cost? There are cloud cost management tools that help allocate and track the costs of running in the cloud, but these are really about gaining visibility into costs, not addressing business value.

94% of organizations adopting Kubernetes say it's a source of pain for their organizations.

SOURCE: [D2IQ, KUBERNETES IN THE ENTERPRISE: UNCOVERING CHALLENGES & OPPORTUNITIES, 2020](#)

Complexity Impacts Business Value



The result of this complexity is, very simply, that your applications don't run as efficiently or optimally as they could. And when engineers are spending thousands of hours annually tweaking, tuning, and troubleshooting at the Kubernetes level, they aren't focused on delivering new capabilities that will deliver business value.

With Kubernetes, You Can't Have it All...Or Can You?

When deploying an application on Kubernetes, there are trade-offs between cost, performance, and time.

- If you want to run an application at a lower cost, you either need to sacrifice performance or spend a lot more time and effort configuring and setting up the app to run more efficiently.
- If you want an app to perform better, you need to accept that it will cost you more or, again, take more time and effort.
- Want to spend less time and effort? Then you need to be willing to have it cost you more or perform worse.

Fortunately, artificial intelligence (AI), specifically machine learning (ML), and automation can help. The trade-offs are still there, but they can be minimized.

AI/ML. AI and ML can go beyond human ability to help find better configurations for applications, which can save money on how applications are run without impacting app performance. It can also improve app performance without adding cost, time or effort.

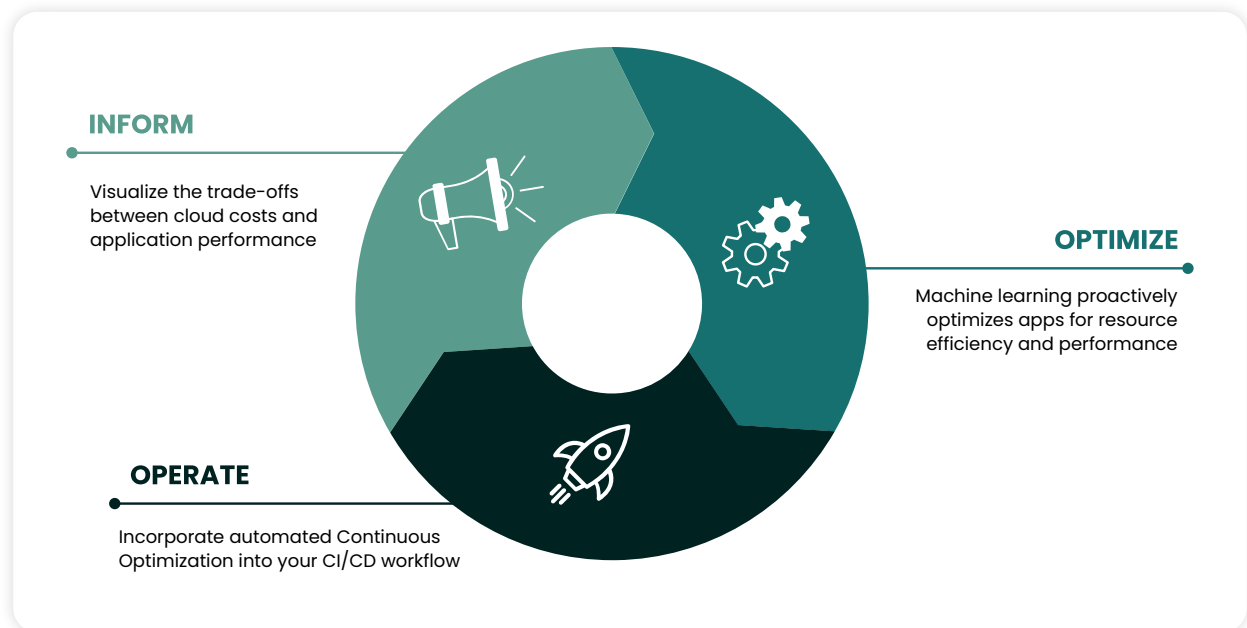
Automation. With automation, engineers can save significant time without increasing the cost of running applications or impacting app performance and reliability.

FinOps: Optimizing for Business Value and Cost

The FinOps lifecycle is a good framework to address the optimization challenges in Kubernetes development environments because it brings financial accountability to the variable spend model of cloud. This framework enables teams to manage the business trade-offs (speed, cost, and quality) necessary to manage cloud costs.

“If it seems that FinOps is about saving money then think again. It's really about making money.”

J.R. STORMENT AND MIKE FULLER, [CLOUD FINOPS](#)



Inform: Visibility & Allocation

Engineers need to understand application efficiency and performance. Gaining visibility helps you see how resource utilization compares to allocation while also providing a better understanding of where cloud spend is going and where that spend should be allocated. It's all about going beyond costs to think about performance and business value.

Optimize: Utilization

Engineering teams need to make smart resource decisions at the container level when configuring a Kubernetes app to effectively run with the right performance at the lowest cost. You can only do this when you understand the trade-offs that have to happen in order to configure applications in the best possible way to meet service level objectives at the lowest possible cost.

Operate: Continuous Improvement and Operations

Continuous optimization needs to be built into CI/CD workflows to create a process that is improving and adapting as the application and environment changes. This requires automating Kubernetes resource management.

StormForge: Shifting Optimization Left

StormForge shifts optimization left by using patent-pending machine learning to automatically find the optimal configurations for applications before deployment. This helps engineers to save time and money while ensuring application performance and resiliency to accelerate innovation.

Purpose built for Kubernetes, StormForge is the only optimization solution that helps you proactively ensure efficiency and intelligent business trade-offs between cost and performance without time-consuming, ineffective trial-and-error.

StormForge Helps Achieve Kubernetes Resource Optimization Goals

Cost Efficiency



Control rising cloud costs, eliminate over-provisioning and cloud waste

Performance



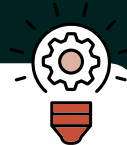
Proactively ensure performance before apps are deployed to production

Automation



Make Kubernetes resource management and optimization a continuous, automatic process

Innovation



Empower development teams to focus on innovation, not tuning, troubleshooting, or going deep into Kubernetes

Automated Kubernetes Resource Efficiency with StormForge

Intelligent Business Decision-making: ML-powered rapid experimentation eliminates time-consuming manual tuning

Using patent-pending machine learning to automatically find the optimal configurations for your applications before deployment, StormForge saves you time and money while ensuring application performance and resiliency. StormForge recommends configurations to minimize wasted resources while meeting performance objectives, giving developers the power to make decisions based upon business goals.

Proactive Resource Optimization: Understand application behavior before deployment

With StormForge, developers can see how applications will behave under realistic load scenarios in pre-production to identify and avoid high-risk configurations likely to result in downtime. And, because our API-first architecture is built for automation and integration into a CI/CD workflow, developers don't need to spend hours on inefficient trial-and-error tuning.

Purpose-built for Kubernetes: Run StormForge anywhere without requiring new resources and expertise

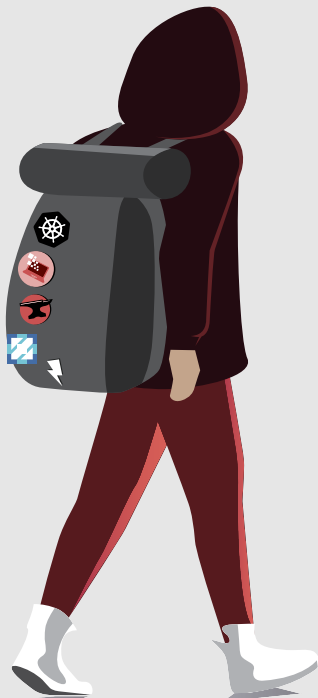
Designed to be cloud and distribution agnostic, StormForge runs anywhere Kubernetes runs, on any CNCF-certified distribution. This means no vendor lock-in and the ability to intelligently scale for more efficient behavior.

Discover the Advantages of StormForge

Let our experts assess how StormForge can deliver the foundation for your Kubernetes success.

Schedule your demo today.

[SCHEDULE DEMO →](#)





+1 (857) 233-9831 | info@stormforge.io | stormforge.io

©2021 StormForge. All rights reserved.