StormForge

# Beyond Cloud Cost Management

## How Kubernetes Optimization Can Help You Do More With Less

# What's Inside

# The Promise of Cloud Native: Move Fast and Scale

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

Organizations move to cloud native for a variety of reasons, including:

- **Agility:** Cloud native architectures enable organizations to rapidly develop, deploy, and iterate on new applications and services, enabling them to respond quickly to changing business needs.

- **Scalability:** Cloud native architectures allow for scaling of applications and services, which means that they can be designed to automatically add or remove resources as demand fluctuates.

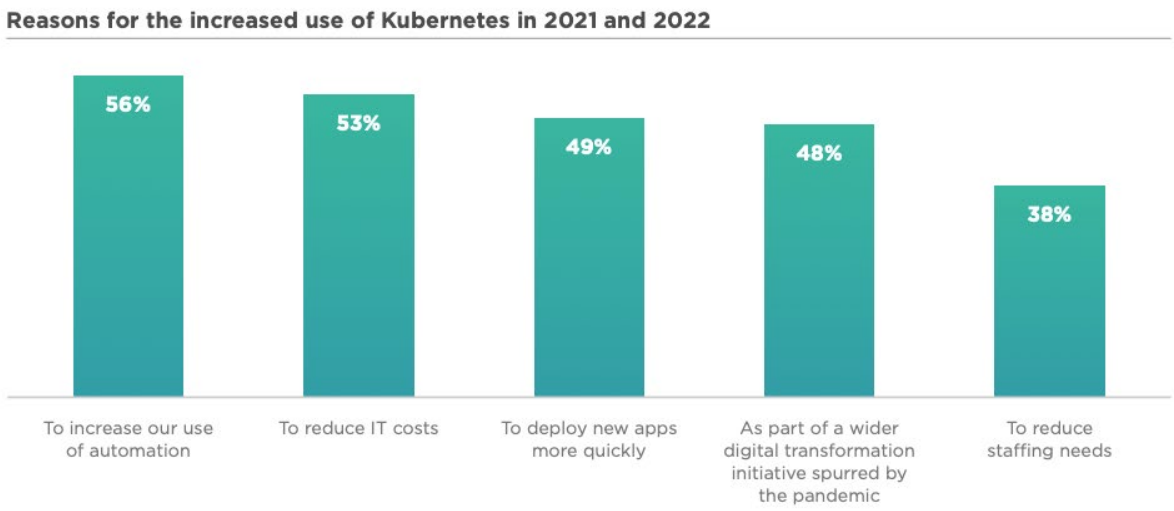
- **Resilience:** Cloud native architectures typically incorporate redundancy and failover mechanisms that enable applications and services to continue operating even if individual components fail.

- **Cost Efficiency:** By leveraging the on-demand nature of cloud computing, cloud native architectures can be designed to only consume the resources that are actually needed, reducing costs compared to traditional infrastructure.

- **Flexibility:** Cloud native architectures allow for easy deployment and management of applications and services across a variety of cloud platforms and infrastructure.

- **Portability:** Cloud native architectures enable applications and services to be easily moved between different cloud platforms, providing greater flexibility and choice for organizations.

As a key enabling technology for cloud native, the reasons for using Kubernetes mirror those for using cloud native.




SOURCE: PORTWORX, 2022 ANNUAL KUBERNETES ADOPTION REPORT

# But is it Really Efficient?

When most organizations start out on their cloud native journey, Kubernetes resource costs and efficiency are not the highest priority. Cost efficiency may be something you hope to achieve, but it's more important to move fast, gain competitive advantage and meet top-line business goals.

It's only after applications are running at scale and day 2 operations become the focus that most teams start to realize costs are higher than expected and still growing.

# Why is this?

1. **Kubernetes is complex.** It's not intuitive or obvious how to best configure Kubernetes resource requests and limits. Decisions are made at a granular level, there are a lot of variables to consider, and the resulting efficiency (performance/cost) is sometimes counterintuitive. Not only that, but usage fluctuates. Autoscaling technologies are meant to help, but they are anything but "auto" - they also require configuration and do not always act intelligently.

2. **Developers are not equipped to make smart business decisions.** For their part, developers are generally not thinking about defining resource requests and limits for Kubernetes apps, and they are rarely empowered to understand the true costs of running applications. In a world where you are being pushed to move fast, what do you do? You guess at resource settings or use defaults, and you definitely err on the side of over-provisioning, because who wants to risk performance and reliability issues for the application you worked so hard to build?

3. **Platform engineers and SREs don't know the needs of the app.** Once an app is in production, it's usually the platform team, SREs or infrastructure/ops teams that need to manage resources. They are also shooting in the dark, guessing at the best resource and autoscaling settings without a complete understanding of the specific application requirements. To make things even more challenging, trying to manage resources at scale across hundreds of services can feel like an impossible task.

**47%** of cloud resources are wasted on average.

**75%** of organizations say their cloud spend is rising.

**70%** say resource optimization and efficiency are high priorities.

SOURCE: STORMFORGE 2022 KUBERNETES & CLOUD WASTE SURVEY

# The New Imperative: Do More With Less

While cloud costs may not have been the priority when we started out, times change and priorities change. Just in the first few weeks of 2023, tens of thousands have been laid off in mass job cuts. The tech sector has been hardest hit, but the reality is that nearly every company is now a tech company at heart, and we're all being asked to do more with less.

If you're working in a role where you build and/or manage cloud native applications, here is your new reality:

- You are being asked (or, more accurately, told) to reduce infrastructure and cloud costs.
- You recognize the complexity of your cloud native environment, but you have fewer people with the skills and expertise needed to effectively and efficiently manage that environment.
- Risking performance or reliability is not an option.
- Slowing down development is also not an option.

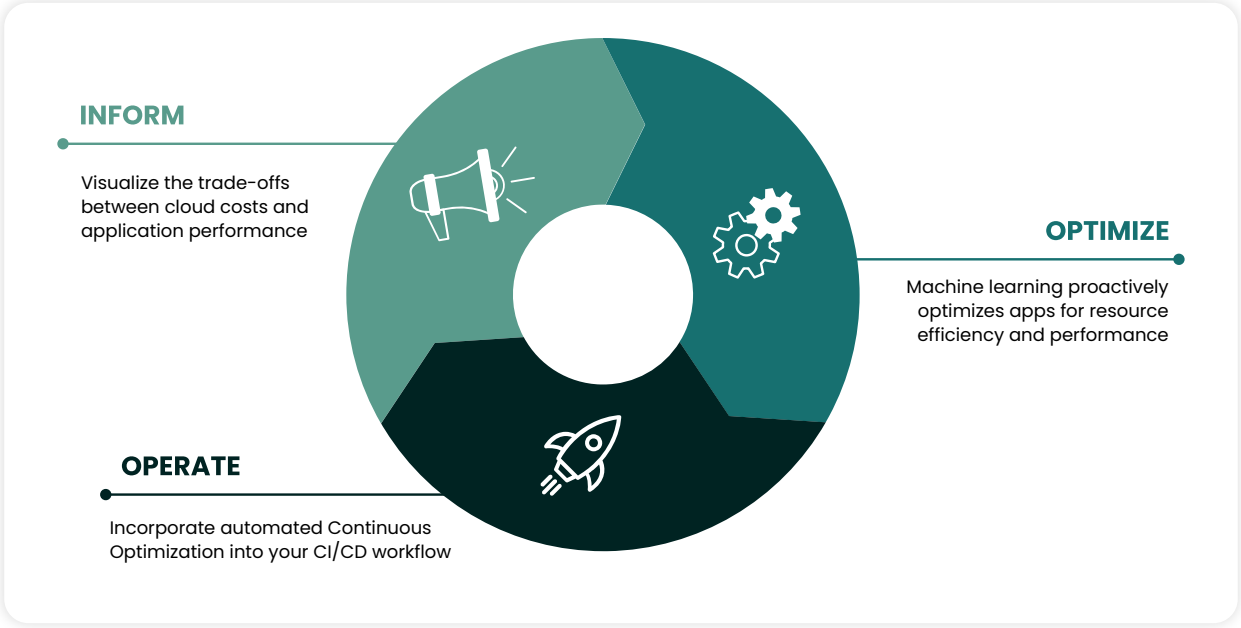
So, where to begin?

# The FinOps Framework

Fortunately, you are not alone. In recent years, new disciplines like FinOps have emerged to help us address these challenges.

As defined by the **FinOps Foundation**:

> *FinOps is an evolving cloud financial management discipline and cultural practice that enables organizations to get maximum business value by helping engineering, finance, technology and business teams to collaborate on data-driven spending decisions.*




SOURCE: ADAPTED FROM THE FINOPS FOUNDATION

The FinOps Framework provides the operating model for how to establish and excel in the practice of FinOps, and includes three phases:

## Inform: Visibility & Allocation

The first step to improving efficiency is getting visibility. Gaining visibility into your environment 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 costs.

## Operate: Continuous Improvement and Operations

Once applications are in production, platform engineering teams and SREs need the ability to monitor, adjust, and improve efficiency on a continuous basis. Efficiency needs to be built into day-to-day operations. Accomplishing this at scale requires Kubernetes resource management that is automated and intelligent.

# Going Beyond Visibility

There are many tools on the market that provide visibility into cloud spend and resource allocation, and that's a good first step. Many of these tools (generally classified as Cloud Cost Management) promise optimization as part of their value. However, the reality is that these tools show you opportunities for efficiency improvements, but they don't tell you what specific actions to take in order to optimize.

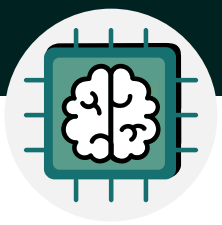
To truly optimize, two additional capabilities are needed: Intelligence and automation.

- **Intelligence** means looking at all the variables that affect efficiency and then providing specific, actionable recommendations on steps you can take to improve efficiency.

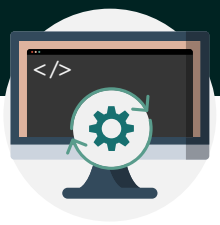
- **Automation** then helps you take the recommended action, and to do this at scale across the hundreds of services running in your environment.

It's important to note also that these capabilities need to be continuous. Optimization is not a one-and-done activity because cloud native environments are highly dynamic and ever-changing.

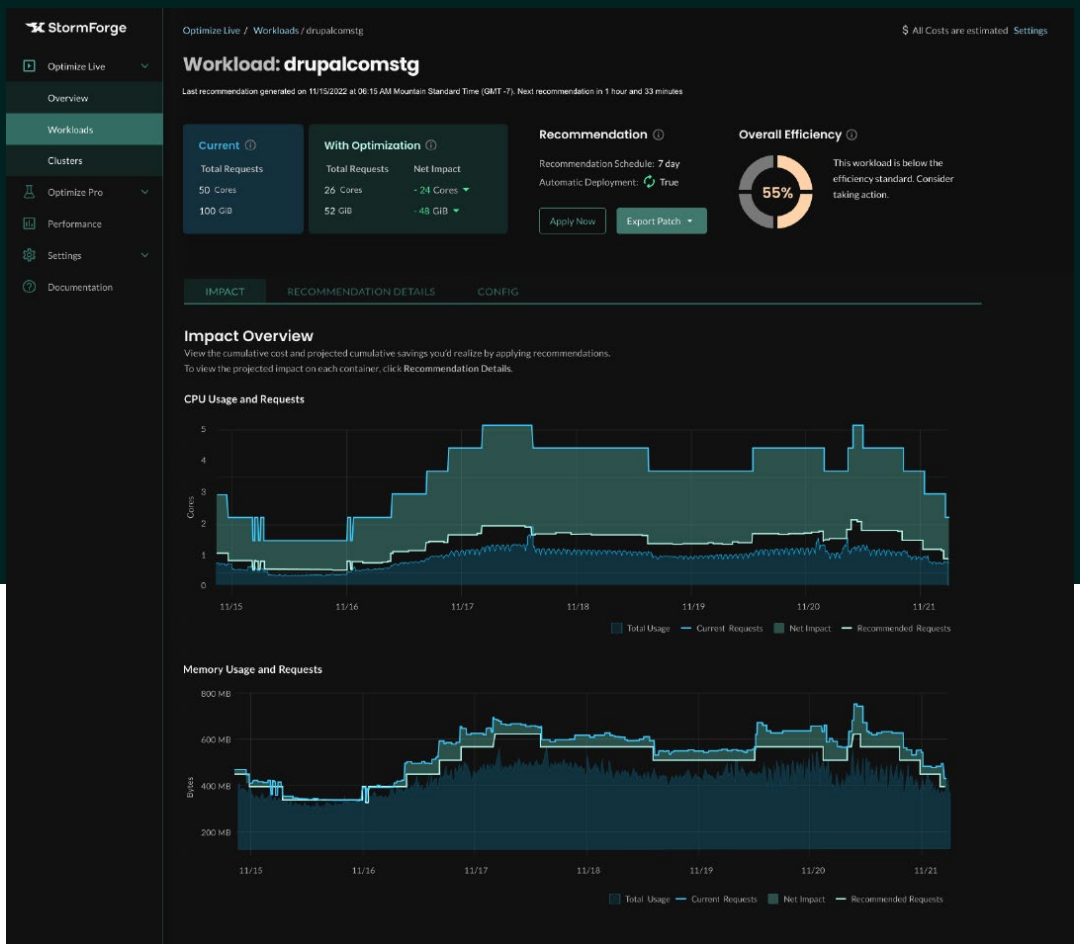
**VISIBILITY**
Show current utilization and identify opportunities for impreovement.



**INTELLIGENCE**
Actionable recommendations to optimize resources as usage varies.




**AUTOMATION**
to proactively and continuously right-size - improving efficiency & eliminating cloud waste.




# StormForge: Intelligent, Automated K8s Optimization

StormForge Optimize Live provides visibility into the efficiency of workloads running on Kubernetes, but it goes beyond that to also provide intelligent, automated action to improve efficiency and keep workloads running optimally on a continuous basis.

StormForge reduces cloud costs and improves reliability by right-sizing your Kubernetes application resources, automatically and continuously. StormForge machine learning analyzes CPU and memory utilization from observability tools like Prometheus and Datadog and automatically adjusts resource requests up or down to meet demand, without risking performance or reliability. Additionally, if you are running the Horizontal Pod Autoscaler (HPA), StormForge recommends and sets the HPA target utilization at the optimal level.

In other words, StormForge does the vertical pod autoscaling for you while also making horizontal pod autoscaling work better and more efficiently.

# Putting the “Auto” in Autoscaling

StormForge Optimize Live is easy to install, configure and use.

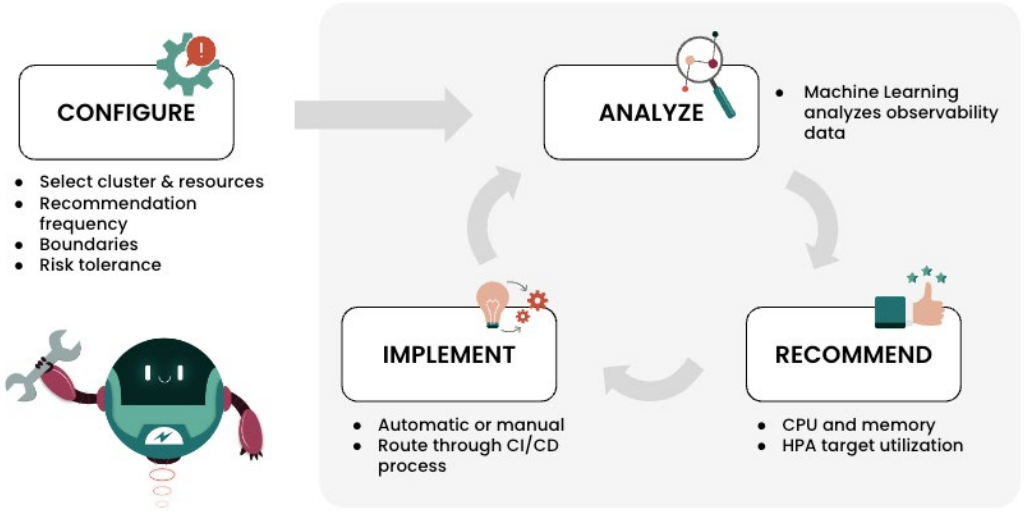
- **Configure.** Configuring StormForge to begin optimizing is simple. Just select your cluster and resources, how frequently you’d like to receive recommendations, and whether you want recommendations to be automatically implemented or if you would prefer to review before implementing. Optionally, you can set boundaries to constrain recommendations and also a risk tolerance to indicate how aggressive you’re willing to get with resources.

- **Analyze.** Next, StormForge machine learning begins analyzing actual resource usage from your observability solution. Once it has seen enough data (usually a few hours worth), StormForge will start making recommendations.

- **Recommend.** Optimize Live will feed you recommendations for updated CPU and memory at whatever frequency you specified during configuration. Additionally, we will automatically detect if an HPA is running for a particular workload. If so, StormForge will also recommend the optimal HPA target utilization.

- **Implement.** If you configured your application for automatic implementation of recommendations, StormForge will automatically patch deployments with the recommended configuration. If you configured your application for manual deployment, you can review the recommendation, including container-level details, before deciding to approve or not.

Additionally, **StormForge works with cluster autoscaling tools like Karpenter** to fully realize cloud cost savings both at the infrastructure and application levels.




# Getting Started

There's no reason not to get started optimizing your Kubernetes resource utilization today.

The benefits are clear:

- Reduce cloud resources by 50% or more on average
- Without sacrificing performance or reliability
- Pods are always right-sized
- HPA always set to optimal target utilization
- Do more with less

**Contact StormForge to get started today.**

> "StormForge has been a game changer for Acquia. It allows us to keep pace in a rapidly moving and highly competitive market, while enabling us to accelerate the value we deliver to customers."
>
> CHARLEY DUBLIN, VP OF PRODUCT MANAGEMENT


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