

# Building a Sustainable Performance & Efficiency Culture



# What's Inside

---

04

Introduction

---

05

Keywords & Concepts

---

06

Creating a Sustainable Path to Better

---

08

Cloud Native Flexibility as a Path for Sustainability - and New Complexity

---

09

Automation Eliminates Developer Frustration and Enables Lasting Gains

---

10

Let a Sustainable Performance and Efficiency Culture Evolve!

---

16

Using Cloud Native for Sustainability Transformation

---

18

Closing Thoughts

## The key message for this ebook:

Building a culture that values performance & efficiency is a smart way to meet the complexity of modern software systems, and also helps enable happy teams and a happy planet.



# Introduction

The various teams within your organization work through different reward systems. Some seek satisfaction in achieving an uptime of +99%, others by squeezing out the last millisecond out of response times. The next team likes to automate as much as possible, and some feel rewarded when a system runs at 30% less costs.

Despite these very different perspectives on cloud-native systems, all stakeholders have at least one thing in common – everyone needs a clear understanding of the system and its behavior. In this eBook, we will describe some of the modern approaches to create these insights and inform a team's culture, and we will propose solutions to invest an organization's resources more sustainably.

- With many departments and stakeholders involved, it is a very complex task for an organization to bring together the potentially competing perspectives on the product.
- Understanding at a component/team level is the basis to achieve a common understanding by all teams/departments of the whole system behavior.
- A common understanding of system behavior is the basis for having valuable business-focused discussions and making the right tradeoffs.
- Seeing infrastructure costs, development velocity & employee satisfaction/happiness as indicators for sustainable application development is a valid approach to modern software delivery.
- We want to strive for "efficient usage of brains." Performing certain aspects of delivering cloud native applications manually is a waste of talent.
- We cannot do better at all of this without planting and growing sustainability thinking and extending our team culture.

With this eBook, we look at how you can take a sustainable path to better resource utilization, developer velocity and happiness, and last, but definitely not least, budget efficiency.

# Keywords & Concepts

## Performance

When addressing a sustainable culture, we don't limit the concept of performance to a software system. Instead, we try to embed it into a broader perspective. In a nutshell: We see performance as the ability for an organization to deliver fast, stable and reliable applications with high velocity while keeping everyone participating in the delivery process in a happy state.

## Cloud Native

When we talk about cloud native we follow the definition given by the Cloud Native Computing Foundation:

*Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*

*These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil. (Source: CNCF, 2018)*

## Optimization

Application optimization is the process of tuning, testing, and re-tuning an application's parameters and configuration settings such that its operational performance is in line with the organization's preferences – whether that be for lowest cost, highest speed, or some other specific parameter.

## Happiness

There is no general definition for the happiness of knowledge workers, but here we follow the idea that everyone smart enough to work on challenging tasks is happy when they're enabled to create a high quality product and learn with and from other people.

## Sustainability

There are many definitions of sustainability, however for this eBook we use a very broad concept of sustainability and refer to it as something with the capability to maintain or "sustain" itself over time. This includes, but is not limited to, the use of natural resources or sustaining our climate.

## Resources

Here, the term "resources" is not limited to computing resources like memory, storage or compute, but also includes human aspects that include time, attention, the happiness of developers, and monetary budget.

# Creating a Sustainable Path to *Better*

Objectives across an organization may differ, but the shared desired outcome doesn't. Despite varying perspectives, diverse stakeholders have at least one thing in common – everyone needs a common and clear understanding of system behavior to provide a basis for valuable discussions and the ability to make the best decisions.

As organizations move to new cloud operating models, adopt tools to build and deploy applications in multi-cloud, and attempt to make better business trade-offs (speed, cost, and quality), they need to take a modern approach to gaining insights, informing a team's culture, and investing resources. This requires a sustainable path to better resource utilization, developer velocity, employee satisfaction, and budget efficiency.

Unfortunately, we can't do better without cultivating and growing sustainability thinking throughout a team culture. And that requires empowering people with the tools they need to gather data and make decisions about cloud native applications and how they are delivered and optimized – in a way that is relevant to each person's perspective.

According to a recent [StormForge survey](#), 48% of cloud spend is wasted on average. [Datadog found similar results](#) looking at container usage in real production environments:

**49%**

of containers use <30%  
of requested CPU.

**45%**

of containers use <30%  
of requested memory.

In 2020, companies wasted over \$17 billion in cloud spend on idle and excess resources.

Source: [DevOps.com, The Cloud Is Booming – But So Is Cloud Waste, March 2020](#)



The StormForge report also revealed that the impact of cloud waste is significant and pervasive, affecting not only IT, but also the business at large, including:

- Reduced profitability
- Poor perception of IT
- Inability to remain competitive
- Difficulty staying within budget

Quality (64%) and to speed deployment of new products and features (53%) are the top two reasons companies decide to implement Kubernetes.

Source: [Study: Circonus Survey: The Complexities of Kubernetes, 2020](#)

# Cloud Native Flexibility as a Path for Sustainability – and New Complexity

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. This flexibility can be the key to a very efficient, automated and effective environment in which to deploy, deliver and operate systems, empowering users to solve for their use case in a highly sustainable way.

At the same time, users are often not thinking about defining resource requests and limits for Kubernetes apps, and they are rarely empowered to understand the true costs of running applications.

Suddenly, this flexibility creates considerable complexity because users don't have the ability to know, when faced with so many options, the ideal configuration to optimize resources and minimize costs. And, without the ability to take ownership of cloud resource usage, and make informed resource decisions quickly, organizations end up spending millions of dollars on wasted cloud resources, suffer business-impacting performance issues, and lose thousands of hours of productivity every year.

# 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](#)



# Automation Eliminates Developer Frustrations and Enables Lasting Gains

By automating repetitive tasks, engineers can save significant time without increasing the cost of running applications or impacting app performance and reliability. Leveraging new tooling always includes effort to learn and operate the solution, but once the initial work is done, automation enables long lasting efficiency gains.

The emergence of open source automation tools such as Ansible (deployment), ChaosMesh (reliability testing), and Kubeaudit (security testing) are already alleviating manual tasks that otherwise would dramatically decrease the benefits of cloud native applications and containers. After all, in order to take advantage of the benefits of orchestration, the tools you use with and for Kubernetes should enable automation.

Optimizing the efficiency of cloud-based applications through a process of rapid experimentation using machine learning and automation can limit the trial and error that comes with manual processes that force engineers to make changes reactively, after an application is in production. With automation, engineers can deliver software fast and reliably with innovative, differentiated capabilities that drive business value (and enjoy the process along the way).

# 51%

of developers and architects say building cloud native applications makes them want to find a new job.

Source: [D2iQ, Kubernetes in the Enterprise: Uncovering Challenges & Opportunities, 2020](#)

# Let a Sustainable Performance and Efficiency Culture Evolve!

No reasonable person wants to build a slow, unscalable and unreliable system, and no rational decision maker would sign a blank check to build and run a system. As developers, we have to deal with limitations and make tradeoffs based on the best information available. This is why it's called software design and not software art. With art, you have much more freedom, but when designing something, you don't necessarily have a blank canvas.

Finding a suitable balance between performance, scalability, reliability and (cost) efficiency is not trivial. However, installing a bunch of tools for testing and monitoring alone does not solve this challenge. IT is more and more in the very center of every business and that means it also has stakeholders across the business, it's a cross-functional endeavour: Development, operations, QA, product, management, marketing and sales, and customer support, to name a few. To solve the balancing challenge, the most sustainable path is to let a performance and efficiency culture evolve among these stakeholders.

The north star of this culture is a common quality pledge. While a cross-department endeavor might sound intimidating at first, there are some proven steps teams can follow to let a sustainable performance and efficiency culture evolve. Let's take a look at the most important steps.

# 01. The Quality & Efficiency Pledge

We motivate our customers and accompany them along the way to shift left testing and optimize early and often. The first step on this is to create a Quality & Efficiency Pledge. This should include commitments for external customers and internal product teams.

## Examples

### EXTERNAL CUSTOMER

*"As a customer, I receive any transaction from the product in <1 second, so that I feel I have a responsive and usable product and stay happy."*

### INTERNAL PRODUCT TEAM

*"As a product company:*

- We commit to our customer quality promise and meet its criteria*
- We make sure that our system resources are in effective use a minimum of 70%\**

*To keep our customers satisfied, be budget efficient, and meet our sustainability goals."*

\**(100% available resources - 30% head room = 70% effective resource usage)*

At the beginning, the Quality & Efficiency Pledge can be simple and become more sophisticated along the way. Once it is in place, teams can continue by defining indicators and metrics to measure progress and success.

## 02. Define Technical Indicators and Metrics

It is a good practice to define (technical) indicators for measuring the current state and future progress.

Define SLOs that are oriented towards the quality & efficiency pledge, track SLIs continuously to understand if a SLO is met or violated. More importantly gain an understanding of potential efforts to be invested to actually meet an SLO. Applying the defined SLOs also in pre-production environments [enables teams to identify risks early on and mitigate them before risks become problems.](#)

## 03. Building the Initial Scope and Heavy Lifting

In the early stages of creating a performance and efficiency culture, it makes sense to include the different stakeholders in initial simple experiments. This gives everyone involved a feeling of participation and can create excitement when experiencing the first results together. We recommend keeping these first experiments simple so that everyone can understand what will be tested or optimized and why and how results can be interpreted as a team with different perspectives.

1. Create first performance test case
2. Guess first numbers within the team
3. Prepare first version of test data
4. Run a small, low traffic test session with the whole team
5. Ask for help within the team and improve iteratively

By having everyone on board for the initial setup, teams can iterate their way closer to real production setups and workloads.

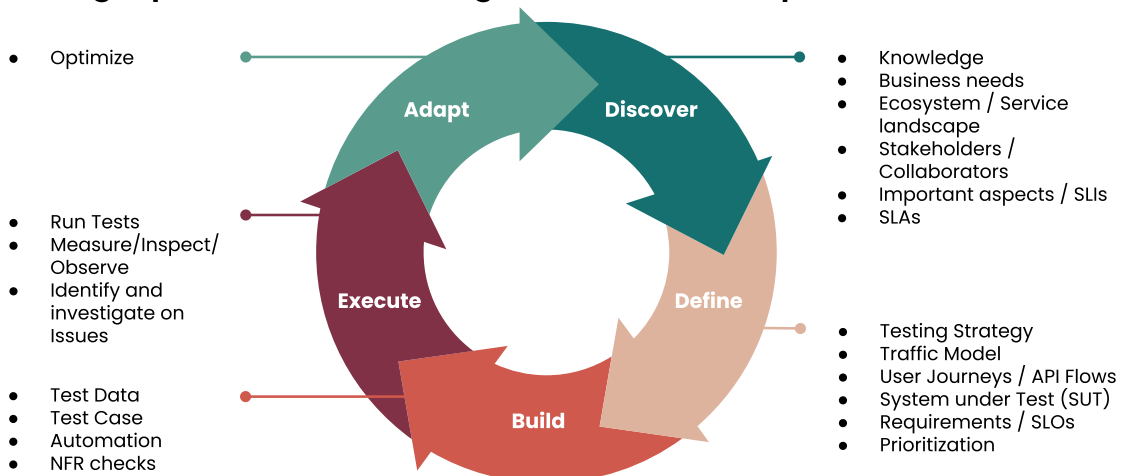
# 04. Iterate & Automate

The next stage of building out your culture involves the software delivery team and is, basically, a normal development motion. Start small and grow iteratively.

When introducing a new feature or endpoint, create a functional test and keep your existing performance and efficiency tests updated and well maintained to allow a consistent and precise measurement against your quality metrics. Commit, integrate, and run your tests regularly.

## The Performance Testing Cycle

Building a performance testing culture and discipline



*With StormForge it is easy to integrate your testing into your CLI and validate non-functional requirements continuously.*

## **05. Over-Communicate and Refine All (Non-Functional) Requirements**

Especially at the early stages of building a sustainable performance and efficiency culture in your organization, it is important to over-communicate. Your Community of Practice should meet regularly and often. This gives everyone on the team the opportunity to repeat the commitment to quality and efficiency and turn it into a vision to strive for.

Every functional team should present results they see within their tools, from metrics to customer feedback, and communicate achievements as well as violations of the defined requirements. Discuss ideas to improve and support each other to remove obstacles. Revise your requirements grounded on regularly communicated results.

# Using Cloud Native For Sustainability Transformation

Most organizations are leveraging the cloud to run their applications, and more and more have [intensified their adoption of cloud native technologies](#). Over the past few years, we can recognize an increase of computing-intensive solutions based on machine learning or IoT, and the [number of people using the internet](#) is growing as well. All these factors increase the energy demand of the cloud. In the near to midterm future, additional technologies like the electrification of transport will put additional stress on the energy supply side and [are already doing that today](#).

Despite the surge of cloud use over the past years, [energy consumption of data centers has not seen linear growth](#) because data center efficiency (servers, cooling) has improved significantly. In the end, data center providers are highly incentivised to innovate on that front as the electricity bill makes up a huge portion of overall costs. Yet, Moore's Law has its limits and we may reach these [sooner than we think](#).



It is important to mention that all major cloud providers do have initiatives to reduce the carbon footprint of their data centers, but the carbon emissions are not the core problem. On one hand [carbon emissions of data centers are very difficult to measure](#), and on the other hand, the energy needs for the cloud don't exist in a vacuum. Other technologies will compete for a limited or fluctuating amount of renewable energy production. Seen from this perspective, it appears to be more important to buy as much time as possible for the large-scale emergence of storage technologies for renewable energy.



**“We used the cloud for digital transformation. Why not use it for a sustainability transformation as well?”**

ADRIAN COCKCROFT, Chief Sustainability Architect, Amazon

[Watch the Cloud Waste Panel 2021 >>](#)

[Adrian Cockcroft](#), the Chief Sustainability Architect of Amazon, stated that “we used the cloud for a digital transformation” and asked “why we are not using the cloud for a sustainability transformation as well”. In December 2021, [AWS published the 6th pillar of their “Well Architected Framework”](#) focusing on sustainability. One of the key thoughts of this pillar is a shared responsibility between cloud providers and users to achieve a sustainable future.

Cloud native technologies like Kubernetes allow a very high degree of flexibility, and therefore, many options to configure for efficiency. Enabling teams to learn how to harness this flexibility and foster a performance culture is not only something that can make a team proud of their creation, but also enables these teams to have a direct impact on a more sustainable future for all of us.

# Closing Thoughts

Building a sustainable performance and efficiency culture will not be easy. But organizations that successfully implement the practices recommended in this eBook will reap significant rewards, including:

- **Business agility** – Organizations with a strong performance culture are able to leverage cloud native technologies to move faster, creating sustainable competitive advantage and driving faster digital transformation.
- **Financial success** – With cloud costs making up an increasing portion of the overall cost of revenues, sometimes [approaching 75-80%](#), responsible allocation and management of compute resources is imperative, and the opportunity for cost savings and increasing company valuations is substantial.
- **User satisfaction** – Organizations that are able to achieve a sustainable performance and efficiency culture see improvements in application performance and stability, the result being happier, more satisfied (and less frustrated) end users.
- **Employee happiness** – While employee satisfaction may seem like a secondary concern, the fact is [employee engagement and happiness are key drivers of customer happiness](#). Especially in a thriving economy with high demand for skilled knowledge workers, employee happiness is critical for retention and for driving digital transformation forward.
- **Environmental responsibility** – While some may think of sustainability as a corporate branding or PR initiative, it's becoming increasingly important to businesses' bottom lines as well. In fact, [92% of consumers are more likely to trust a company that supports social or environmental issues](#).

As leaders in our respective organizations, it's our responsibility to create an environment where people can thrive and where performance and efficiency are ingrained in the culture. The steps outlined in this eBook are a good start to set you on the road to success.

# Discover the Advantages of StormForge

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

Schedule your demo today.

[REQUEST A DEMO →](#)





+1 (857) 233-9831 | [info@stormforge.io](mailto:info@stormforge.io) | [stormforge.io](https://stormforge.io)

©2021 StormForge. All rights reserved.