



Survival Guide:

Nine Tips to Recession-Proof Your Infrastructure

The AWS logo, consisting of the lowercase letters 'aws' in a white, sans-serif font, with a curved orange line underneath that resembles a smile or a checkmark.

PARTNER

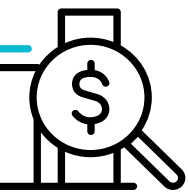


Survival Guide:

Nine Tips to Recession-Proof Your Infrastructure

Uncertainty is in the air. Due to economic turmoil and inflationary pressures, many businesses are preparing for the worst and a potentially protracted downturn. This puts the squeeze on budgets and on staff to continue delivering core services and doing more with less. To survive and thrive during these difficult times, every prudent business should be thinking through and preparing to make some hard decisions regarding optimization.

If you are in charge of running cloud infrastructure for your organization, you may be exploring ways to ensure you can support shifting business priorities while meeting cost-cutting goals. This guide will present nine practical ideas you can implement to recession-proof your infrastructure.



1. Tighten the Belt on Your Biggest Workloads

Most workloads are overprovisioned with computing resources, especially when they are not running on elastic infrastructure that can scale up and down to match demand. Optimizing workload performance means spending with higher accuracy while not sacrificing performance and availability. Enterprises that accurately provision infrastructure and engineering resources can successfully optimize their more critical and expensive work-

loads with confidence. To find big savings, you have to look at the biggest workloads; small economies here can lead to the largest gains.

“You need a way to protect your customer experience, ensure that cloud services perform as expected, and cost-tune your new hosting architecture so that it works well and is cost-efficient within well-defined goals.”

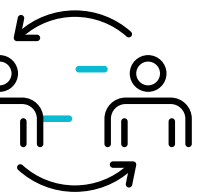
Source:
[Moving to the Cloud? SLOs Will Help](#)



2. Consolidate Spend with Managed Open Source

Infrastructure, DevOps, site reliability engineering, and observability tools can be expensive, but robust, well-supported open source alternatives are often available. For example, you may be able to reduce or eliminate expensive legacy tools and replace them with easy-to-use AWS services like Amazon Managed Service for Prometheus (AMP), saving as much as 80% on fees.

Further Reading:
[SLOs Made Easier with Nobl9 and CloudWatch Metrics Insights](#)



3. Share the Load

Your team may be broken into specialist groups like development and operations. You may have overprovisioned your compute infrastructure by using dedicated infrastructure for different services and applications. Consider how you might be able to promote cooperation between teams and roles to improve operational efficiency and cost management. You may have to slow down on feature delivery to operate the service, but the “you build it, you run it” philosophy frequently leads to better results through

clear incentives anyway. Compressing roles and capacity can free up resources, allowing you to reduce spending or work on other critical areas that are being overlooked. Mind the gaps, and tighten them up. Business requirements may not be aligned with IT objectives; for example, IT may think the business requirement is five nines when really it's below that, and have overprovisioned infrastructure.

“Run what you build forces the entire IT team to understand more about the customer. That knowledge will no longer be limited to a product or sales team, and these insights can be incredibly useful when used as a feedback loop for constant product improvement.”

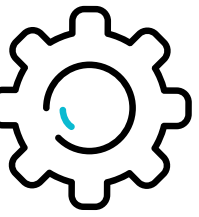
Source:
[Why You Should Run What You Build](#)



4. Follow the Money

Cash is the lifeblood of your business, so you need to know where it's coming from and where it's going. As the owner of the infrastructure, you need to make sure you align your cost savings to your business priorities – which may have changed overnight. For example, your organization may be re-focusing on safer core business instead of higher-risk growth or expansion areas. “Next-gen” infrastructure platforms often require huge investment before they generate cash, and priorities may have shifted from trying to expand into new areas to shoring up existing cash generators. Understanding this shift can allow you to quickly cut expenses in areas that are still early in their product development cycle. Make sure every IT project has a clear business “why”.

Further Reading:
[Measuring Technology ROI: SLOs for CFOs](#)



5. Rightsize Redundancy

Reliable systems inherently require redundancy to tolerate faults and other disruptive risks. However, not all parts of your system are mission critical (especially during recessionary times), and these may be able to tolerate the additional risk of downtime. It's best to go into this knowingly and make explicit decisions about how much risk is acceptable for the various services, and to define these goals in an agreed format like a Service Level Objective (SLO). By setting lower reliability targets for less critical systems, you can free up valuable resources to reinvest into the most critical ones. Recognize that not all services are equal and that you will have to target different levels of reliability for different use cases, regions, and user groups.

“Having a limited amount of downtime within a timeframe isn't fully unexpected behavior, but using SLOs will help us manage how much of an impact we are impressing on our customers from that downtime.”

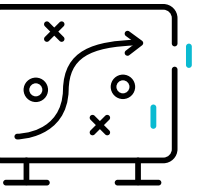
Source:
[How to Convince Anyone to Adopt SLOs](#)



6. Clean Up Storage Space

Storage tends to grow and grow. In the infrastructure space, a big part of the problem is the accumulation of monitoring, observability, and log data that will never be looked at again. It's collected and stored “just in case,” but the diagnostic value of this data falls off sharply after a few hours. Collect data directly related to business KPIs, and store this small set for the long term. Reduce collection of monitoring data to action-oriented data with a short shelf life, and resist the temptation to store everything. Setting SLOs tied to business outcomes will help you focus on metrics that matter to users and business goals and not merely implementation-specific infrastructure concerns.

Further Reading:
[You Already Have Enough Data for Good SLOs](#)

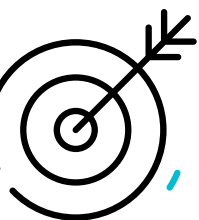


7. Lifeboat Test Your User Journeys

Generally speaking, you want your users to have a great experience no matter what they are trying to do. However, some experiences – like the first time they see your app – matter much more to the business than others. Strictly prioritizing user journeys in terms of frequency, retention risk, and revenue value and setting clear and actionable goals (with automated notifications and alerts) will bring focus to the most critical user experiences. You can then potentially trim infrastructure and engineering spending on lower-priority user journeys. Your users may experience small doses of frustration on rare occasions, but this approach results in a system that is more efficient overall and bulletproof in most critical cases.

“The goal is to tune your service metrics so that they have a high correlation with user happiness, which will allow your teams to trust them when they are trying to optimize the reliability of a system or when they are stable and can now focus on improving the user experience and adding features.”

Source:
[How Do We Measure the Customer Experience?](#)



8. Set Reasonable Goals

Making dramatic cuts and killing expensive projects with one stroke may be warranted if you have a lot of bloat and Skunk Works projects without a clear purpose. More likely, you already run a tight ship and you need to cut with a scalpel, not a chainsaw. Latency goals should be driven by user expectations. By defining sensible reliability targets, allowing some tolerance for risk, and having an adequate response plan, you can make huge efficiency gains across your projects without your users even noticing! Having reasonable goals nudges the organization toward more efficient use of

resources and eliminates the stress from absurd unfunded mandates.

Further Reading:

[The Edge of Excellence: How to Delight Customers at Scale in Digital Services](#)

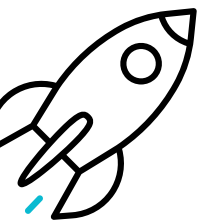


9. Innovate Judiciously

You can't put a halt to innovation, or you'll risk losing out to the ferocious competition. However, you can set constraints around what new innovation – particularly related to long-term bets like Web3 and AI/ML – is prioritized during this period of uncertainty. Cap the infrastructure spend for non-revenue-generating projects and prioritize by likelihood of success. You will likely be able to double down on innovation in the future, but for now, it's better to set clear constraints in alignment with business priorities.

Further Reading:

[How to Keep Your Developers from Idling in Tech Debt](#)



Be Ready for the Comeback

Nothing lasts forever, and that includes the current economic conditions. When the time is right, you want to be poised for a comeback with a lean operating environment. Running efficiently is important at all times, and it's much easier to stay fit than to get in shape. By analyzing your environment and making fast adjustments, you can recession-proof your infrastructure and ensure that you're ready to ramp up again as soon as it's practical to do so.

