nobl9.com

Do You Really Need Five Nines? - Nobl9

Kit Merker

7-9 minutes

Call it what you will: Always On, Six Sigma, High Availability, or Five Nines. Management wants you to deliver a service as close to perfection as possible. Little do they know about the downsides of high availability. In a world where applications are delivered over public networks, the physics of only five minutes of downtime per year is an expensive longshot at best. How do you actually achieve five-nines when you need it? And how do you know when you don't need it? This post explains why, and how your teams can develop a common understanding of availability that focuses on the customer experience, not arbitrary availability metrics.

What does it take to keep customers happy?

Today, most people expect cloud services to just work. We are happier when applications and services are always available than when "the network (or server) is down." A corollary more to the point: Unavailable apps equals unhappy customers—customers can't use a service that's down. Even small glitches in an available service are extremely problematic. What good is it if I can reach my bank's app and check depositing won't work?

> Rather than aiming for too much of a good thing, let's prioritize our services and choose carefully the level of availability each service needs.

Availability is the very definition of competitive advantage in cloud computing. Some cloud providers even say the availability of their network is the most important facet of their services. Enterprise infrastructure teams make the same argument. That's why you hear so many people throwing around the term "five nines" these days. Five-nines availability means your service is available 99.999% of the time (hint: count the nines). In other words, a system with five-nines availability is *unavailable* only 0.001% of the time, which means errors are expected to occur only *5.26 minutes per year*. That's an impressively reliable system (and an expensive one)!

To be sure, customers should be quite happy with five-nines availability, so setting that as a goal makes good sense. . . or does it? Does it really take five nines to keep your customers delighted with your service, especially when you look at their journey through your application? The question is an important one, for one very simple reason: cost.

Delivering five-nines availability is expensive. Each nine added to your availability percentage costs your organization an order of magnitude more. For a customer-facing service to be five-nines available, all the supporting services have to have *even higher* levels of availability. You can think of it this way: every nine adds another zero to the end of your total cost of ownership (people and infrastructure). Hypothetically, if you wanted to run ALL of your services at five nines, your budget would swell from $1M to $10M. Is it worth it? Probably not, especially since the more realistic outcome of trying to achieve five-nines is failure. You may think you are prepared to deliver high-availability service, but, in reality, availability at that level would just be a lucky coincidence.

For most of your services, the chances are customers won't notice the difference between three or four nines and five nines. In fact, three nines is an impressive result that takes thoughtful planning and investment. In the majority of cases, trying to achieve the

extremely high availability above that isn't fiscally justified: it will cost you too much money and too much time, and you'll be wasting resources where they aren't needed. On the other hand, most businesses have a minority of cases where high-availability *is* justified. Focus your investment on those critical services.

So, let's talk about a smarter approach:

Rather than aiming for too much of a good thing, let's prioritize our services and choose carefully the level of availability each service needs. We'll call these targets our "Service Level Objectives" or "SLOs" for short.

- What are the top 2-3 use cases that are mission-critical? For these services, you may opt for an SLO of four or five nines.
- What services are obviously of lower priority? Here, SLOs of three nines or four nines are likely sufficient.

  If you are an e-commerce site, for example, you may place a higher priority on the checkout experience than simply browsing your catalog, choosing an SLO of five nines for purchasing, and four nines for browsing. Or perhaps you have a development environment with an SLO of three or four nines and a production environment with an SLO of five nines.

  As you see, you may end up with a mix of SLO targets ranging from three to five nines. You're keeping customers happy without wasting time and money trying to achieve a level of performance that doesn't deliver commensurate benefits. You've also determined the critical services that are worth extra investment for automation, redundancy, testing, and other reliability enhancements. That's a far wiser approach than arbitrarily setting SLOs of five nines for everything.

Achievable Goals vs. Aspirational Goals

Unfortunately, you can't just flip the switch by setting reliability targets. The reliability goals you set may be your aspiration, but you have hard work to do to achieve these goals. You may want to benchmark yourself today and look at your historical track record with these metrics. Catalog the risks that may affect your service—external outages, software updates, network congestion, etc.—and try to quantify their impact in terms of reliability. For each, calculate the estimated frequency, time to detect and resolve, and impact on the user base. This gives you an idea of expected downtime per year per identified risk. By adding these up, you can create a model for the achievability of your goal.

Once you've set the benchmark, you'll see a clear gap between today's risk profile and your desire to deliver a more reliable service. Starting from your biggest risk, ask your team, "How could we proactively mitigate this risk?" For example, you might find a single point of failure that needs redundancy. Each of these mitigations can be listed on your reliability roadmap and sorted into the rest of the feature priorities for the release. Bubbling up specific quantifiable reliability improvements is a great way to get them funded.

So, choose wisely

More nines is all the rage, but don't jump on the bandwagon without thinking it through, because as you pour your resources into achieving higher reliability, you will inevitably reach a point of diminishing returns, and passing that point is a waste of your company's precious resources. SRE, after all, is about achieving balance given a limited set of resources.

The better way to set SLOs for availability—or any other critical service objective such as latency, durability, etc.—is to remember it's not about more nines; it's about setting the right number of nines to deliver a great customer experience. To execute this smarter

approach, first, determine which services deserve five-nines; choose wisely, and invest your energy and resources there. Second, define clear, reasonable SLOs for *all* services to make sure your team knows how to prioritize their time.

If you're ready to dig a bit deeper into setting SLOs, check out the next blog post, Optimizing Cloud Costs through Service Level Definitions.

Image Credit: Takahiro Sakamoto on Unsplash