

O'REILLY®



97 Things Every Cloud Engineer Should Know



Edited by Emily Freeman & Nathen Harvey

The Basics of Service-Level Objectives

*Kit Merker,
Brian Singer,
and Alex Nauda*
Nob/9



“I want it to work perfectly,” your boss or maybe a product manager tells you. But you, as a cloud engineer, know they aren’t really willing to pay for that level of service, even if it were possible, which it’s not.

How can you give management an easy way to instantly understand the trade-offs between reliability, speed of innovation, and cost? Service-level objectives (SLOs) are the answer. SLOs create clear reliability guidelines that balance the trade-offs between cloud costs, speed of change, and external risks.

What Are SLOs?

SLOs are key performance indicators for cloud services *based on customer happiness*. SLOs define the precise level of service that needs to be achieved in order to avoid unacceptable risk of displeasing the customer.

Let’s use availability as an example. When we talk about how often our infrastructure is available (uptime), we typically speak in terms of *nines*. If your infrastructure is available *four nines*, or 99.99%, it will be unavailable 52.6 minutes a year. However, if your infrastructure achieves *five nines*, then your system is up and working 99.999% of the time—that is, it’s down only 5.26 minutes a year.

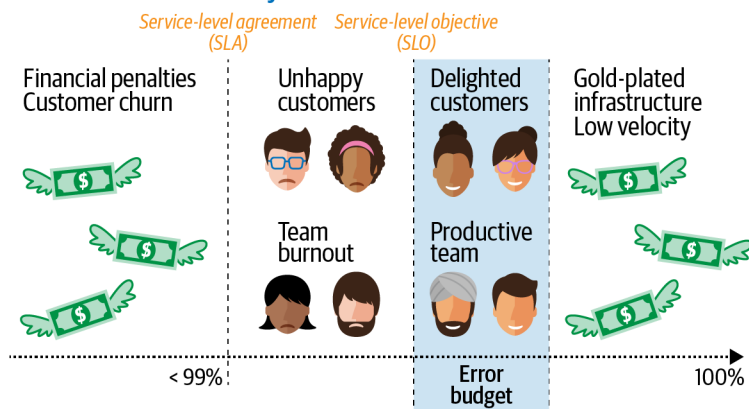
In an ideal world, we’d want our infrastructure to achieve as many nines as possible; however, moving from one class of nines to the next-higher class is roughly 10 times more expensive (you’ll incur significant people and infrastructure costs to make the leap to the next level). And when you consider the inherent limitations of physics and the architecture of public networks, approaching five nines of reliability consistently can become nearly impossible. So, the key question is, how many nines are good enough to keep the customer happy without wasting resources?

Another way of looking at SLOs is to consider their corresponding *error budget*, a small acceptable allowance for errors. If our SLO is 99.99% uptime, our error budget is the additive inverse: $1 - 99.99\% = 0.01\%$, or 52.6 minutes a year. In other words, we will tolerate errors within this allowance—this small amount of downtime—because we expect that outages within this range will not upset customers enough to warrant prevention.

SLOs: The Cloud Engineer's Best Friend

SLOs help bring organizations together around reliability. You're not just chasing nines for nines' sake, pretending to be perfect, or hoping dumb luck is on your side.

SLOs balance reliability and innovation



Here are three ways SLOs can be a cloud engineer's best friend:

SLOs help you collect data about how well you are serving your customers.

Achieve your SLOs, and customers remain happy; blow your error budgets, and customers leave.

SLOs enable you to evaluate business trade-offs.

SLOs help you balance the competing interests of product stakeholders who want to rapidly launch new features or products, and IT operators who want to maximize infrastructure performance and reliability.

SLOs serve as a common language between business and multiple technical stakeholders.

SLOs help everyone in the company work together as an aligned team to delight users and grow the business—and do it without guessing.

Where Do You Start?

As in the preceding example, availability of cloud services is a great place to start. You may also want SLOs to address quality, throughput, latency, and more. In fact, you will eventually have not one but many SLOs, because a variety of things to be measured play a key role in delivering customer happiness.

To get started, choose the cloud service aspect that is most critical to your business. Then, as your teams learn what SLOs are all about and what a great tool they are for balancing reliability, cost, and innovation, broaden your scope to include SLOs for other aspects of your cloud services, such as automation, infrastructure, applications, and user journeys.

As you expand the reach of SLOs in your organization, remember: SLOs must always be directly connected to customer happiness and business impact.

O'REILLY®

97 Things Every Cloud Engineer Should Know

If you create, manage, operate, or configure systems running in the cloud, you're a cloud engineer—even if you work as a system administrator, software developer, data scientist, or site reliability engineer. With this book, professionals from around the world provide valuable insight into today's cloud engineering role.

These concise articles explore the entire cloud computing experience, including fundamentals, architecture, and migration. You'll delve into security and compliance, operations and reliability, and software development. And examine networking, organizational culture, and more. You're sure to find 1, 2, or 97 things that inspire you to dig deeper and expand your own career.

Emily Freeman is a technologist and a storyteller who helps engineering teams improve their velocity. As the author of *DevOps for Dummies*, she believes the biggest challenges facing developers aren't technical but human. Her mission is to transform technology organizations by creating company cultures in which diverse, collaborative teams can thrive. Emily is a principal cloud advocate at Microsoft.

Nathen Harvey is a cloud developer advocate at Google, helping the community understand and apply DevOps and SRE practices in the cloud. He's part of the devopsdays global organizing committee and was a technical reviewer for the *Accelerate State of DevOps Report*. Nathen formerly led the Chef community, cohosted the *Food Fight Show*, and managed operations and infrastructure for a diverse range of web applications.

CLOUD

US \$44.99

CAN \$59.99

ISBN: 978-1-492-07673-5



Twitter: @oreillymedia
facebook.com/oreilly

Three Keys
to Making the
Right Multicloud
Decisions

—Brendan O'Leary

Serverless
Bad Practices

—Manasés Jesús
Galindo Bello

Failing a Cloud
Migration

—Lee Atchison

Treat Your Cloud
Environment as
If It Were
On Premises

—Ilyana Garry

What Is Toil,
& Why Are
SREs Obsessed
with It?

—Zachary Nickens

Lean QA:
The QA Evolving
in the DevOps
World

—Theresa Neate

How Economies
of Scale Work
in the Cloud

—Jon Moore