PLATFORM · TEMPLATE

SLOs for Small Teams: Error *Budgets* Without a Platform Team

KAANSYSTEMS.COM/LIBRARY/SLOS-FOR-SMALL-TEAMS · SEPTEMBER 16, 2026

ABOUT THIS TEMPLATE

SLOs for small teams are 1-3 targets on your most user-facing service, measured from the user's side, with an error budget you spend to decide when to slow down.

THE TEMPLATE

A starter SLO worksheet. Fill one block per SLO; a small team has one to three of these total, not one per service. Publish it where on-call can find it, and revisit it quarterly.

One block per SLO (aim for 1-3 total)

- Service: _____ (your single most user-facing service first)
- SLI — what you measure, from the user's side: _____
- Measured where: ☐ load balancer / ingress ☐ synthetic check ☐ _____ (not host CPU, not RPS)
- Target (SLO): _____% over a 28-day rolling window (and/or p95 < _____ ms)
- Error budget: 100% − SLO = _____ (e.g. 99.9% → 0.1% ≈ 40 min / 28 days)
- Burn-rate alert fires when: on track to exhaust the budget before the window resets — NOT on every dip
- When the budget is HEALTHY (> ____% left): ☐ ship normally ☐ canaries auto-promote ☐ migrations proceed
- When the budget is BURNING / EXHAUSTED: ☐ freeze risky deploys ☐ no migrations ☐ on-call pivots to reliability
- Owner — who watches this number: _____
- Reviewed quarterly: ☐

Two rules that keep the worksheet alive rather than decorative:

- **Start with one.** The most user-facing service, one success-rate SLO, one error budget. Add the second only after the first has changed a real decision. A single working SLO beats four aspirational ones.
- **If it never changes a decision, delete it.** An SLO that never freezes a deploy or greenlights a risk is not measuring anything you use. Cut it, and put the attention on the number that does.

Get one SLO to the point where a burning budget actually slows the team down and a healthy one speeds it up, and you have the whole discipline — the same loop the GitOps rollout runs automatically and the on-call system leans on, minus the platform team you were told you needed.

WHAT ARE SLOS, AND DO SMALL TEAMS NEED THEM?

Yes — a light version, and the concept matters more than the machinery. Three terms, in order:

- **SLI (Service Level Indicator):** something you measure about the service from the user's perspective. The share of requests that succeed. The p95 request latency. A measured number, nothing more.
- **SLO (Service Level Objective):** a target for that indicator over a window. "99.9% of requests succeed, and p95 latency stays under 300ms, over 28 rolling days." The SLI is the number; the SLO is the line you draw across it.
- **Error budget:** 1 minus the SLO. At a 99.9% target, 0.1% of requests are *allowed* to fail. That allowance is not a failure — it is a resource you get to spend on risk.

Small teams need the concept, not the organization. The two failure modes are equal and opposite. Skip SLOs entirely and on-call pages on noise, because nobody ever defined what "broken" means, so every blip is a judgment call at 2 a.m. Import the full enterprise SRE apparatus and you get an SLO for every microservice, a wall of dashboards, and a ritual nobody consults. The middle — one to three SLOs on the service that matters, wired into how you deploy and how you page — is the whole win, and it is reachable this quarter.