Better Reliability With SLOs

Daniel Maher
Developer Relations

DATADOG

Better Reliability With SLOs

Daniel “phrawzty” Maher
Developer Relations

DATADOG

What we'll cover today

- SLIs, SLOs, and SLAs
- Defining quality targets
- Error budgets
- Practical examples

SLIs, SLOs, & SLAs

@phrawzty

SLIs

A Service Level Indicator is a quantitative measurement that expresses an aspect of the service (commonly a metric).

Image credit: Indicator by Nhor from the Noun Project

SLOs

A Service Level Objective is a target value for a service, as measured via an SLI.

Image credit: Objective by priyanka from the Noun Project

SLAs

A Service Level Agreement is a contract that defines the results (and consequences) of meeting (or missing) one or more SLOs.

Image credit: Checked document by Rockicon from the Noun Project

Multiple Stakeholders

Product Managers

Developers, SREs

Executives

Customers

Focus on user experience

SLIs

SLOs

SLAs

Defining quality targets

User experience is everything

- How are they interacting with your product?

User experience is everything

- How are they interacting with your product?
- What is their workflow?

User experience is everything

- How are they interacting with your product?
- What is their workflow?
- What services do they interact with?

User experience is everything

- How are they interacting with your product?
- What is their workflow?
- What services do they interact with?
- What do they want? What do they expect?

Not all values make good SLIs

- Free resources (CPU, Memory, Disk Space)
- Quorum state (does the leader matter?)
- Number of lines of code per commit.

Identifying good SLIs

Response/Request	**Availability** – Could the server respond to the request? **Latency** – How long did it take for the server to respond to the request? **Throughput** – How many requests can be handled?
Storage	**Availability** – Can the data be accessed on demand? **Latency** – How long does it take to read or write data? **Durability** – Is the data still there when it is needed?
Pipeline	**Correctness** – Was the right data returned? **Freshness** – How long does it take for new data or processed results to appear?

SLIs are applied values

Indicators must have represent *user experience*.

The number of requests to an endpoint that complete successfully.

The number of requests to an endpoint that complete within 500ms.

SLOs are applied SLIs

Objectives have both a *target* and a *time window*.

Requests are 99.95% successful in the last 24 hours.

90% of requests complete under 500ms in the 30 days.

SLAs are applied SLOs

Agreements address *expectations* and *impacts*.

The customer expects a given service to have a 0.05% maximum error rate daily, or they'll receive a rebate.

The customer expects only 10% of monthly requests to take longer than 500ms to complete, or they'll be reimbursed for the compute overage.

Error Budgets

Move fast and *fix* things!

- Failure is unavoidable; how you respond is important.
- Balance innovation and novelty with reliability and stability.
- Similar to an SLA.

Building an Error Budget

- An SLO is identified by the product owner.

Building an Error Budget

- An SLO is identified by the product owner.
- The *actual* objective is measured by a neutral party (hint: a monitoring system).

Building an Error Budget

- An SLO is identified by the product owner.
- The *actual* objective is measured by a neutral party (hint: a monitoring system).
- The difference between the actual measurement and the objective is the error budget.

Using the budget

Spend the budget

If the SLO is currently being met, you have room to move. Add new features! Deploy a new version! Trigger some planned downtime…

Build the budget

If the budget is zero (or negative), you should concentrate on that. Freeze new features. Improve the observability story. Prioritise dealing with technical debt.

Practical examples

fotosite.neð

(not real)

- A fun site for uploading, sharing, and viewing photos.
- Make friends! Build communities!
- Buy print-quality photos.

@phrawzty

Consider the user experience

- *How are they interacting with your product?*
  - Clicking, viewing photos, doing searches.

Consider the user experience

- *How are they interacting with your product?*
  - Clicking, viewing photos, doing searches.
- *What is their workflow?*
  - Log in, search photos, view, download.

Consider the user experience

- *How are they interacting with your product?*
  - Clicking, viewing photos, doing searches.
- *What is their workflow?*
  - Log in, search photos, view, download.
- *What services do they interact with?*
  - Accounts backend, upload service

Consider the user experience

- *How are they interacting with your product?*
  - Clicking, viewing photos, doing searches.
- *What is their workflow?*
  - Log in, search photos, view, download.
- *What services do they interact with?*
  - Accounts backend, upload service
- *What do they want? What do they expect?*
  - They want it to be fast!

fotosite.neð: Indicators

Response/Request	**Availability** – Could the server respond to the request? **Latency** – How long did it take for the server to respond to the request? **Throughput** – How many requests can be handled?
Storage	**Availability** – Can the data be accessed on demand? **Latency** – How long does it take to read or write data? **Durability** – Is the data still there when it is needed?
Pipeline	**Correctness** – Was the right data returned? **Freshness** – How long does it take for new data or processed results to appear?

fotosite.neð: Indicators

Response/Request	**Availability** – Could the server respond to the request? **Latency** – How long did it take for the server to respond to the request? **Throughput** – How many requests can be handled?
Storage	**Availability** – Can the data be accessed on demand? **Latency** – How long does it take to read or write data? **Durability** – Is the data still there when it is needed?
Pipeline	**Correctness** – Was the right data returned? **Freshness** – How long does it take for new data or processed results to appear?

fotosite.neð: Indicators

Response/Request	**Availability** – Could the server respond to the request? **Latency** – How long did it take for the server to respond to the request? **Throughput** – How many requests can be handled?
Storage	**Availability** – Can the data be accessed on demand? **Latency** – How long does it take to read or write data? **Durability** – Is the data still there when it is needed?
Pipeline	**Correctness** – Was the right data returned? **Freshness** – How long does it take for new data or processed results to appear?

Bonus: More tips!

Measuring SLIs with Datadog

Monitor-based

- Based on *monitors*, which are generally tied to metrics.
- Values within a set time frame.
- ex. “99% of the time, latency for this request is less than 200ms”

Event-based

- Based on *events*, which are more akin to statements.
- Effectively a success ratio.
- ex. “99% of requests have latency less than 200ms”

Four Golden Signals

Latency

Traffic

Errors

Saturation

Focus on your users.

- Step back from the internals.
- Define user stories / journeys *first*, then SLIs.
- Involve all stakeholders, especially product.

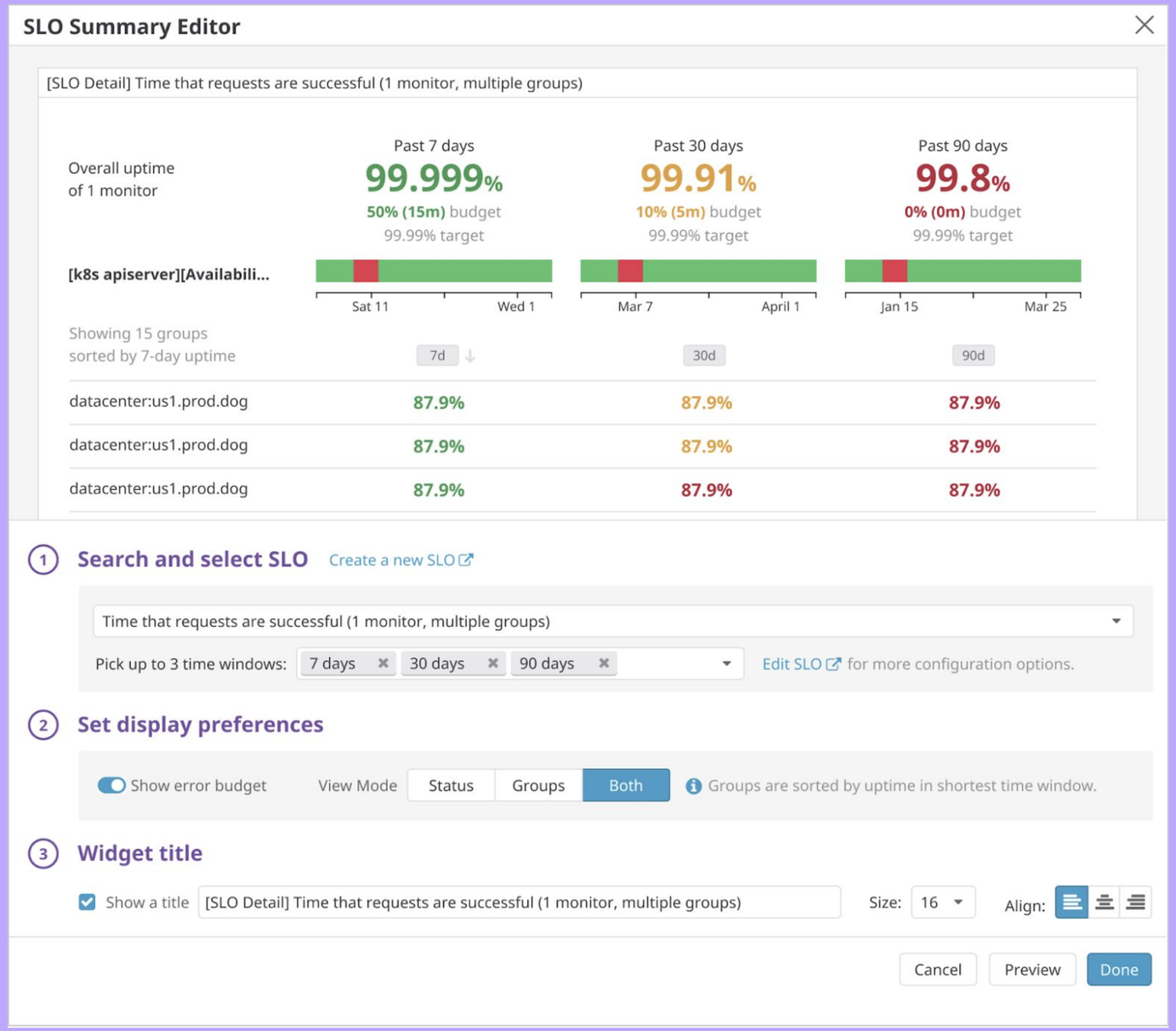
Start small.

- Gain experience; experiment by hand if that's easier!
- Build out data sets to establish reasonable baselines.
- Error on the side of naïveté (at first).

SLOs change.

- Re-evaluate your SLIs and SLOs as your environment evolves.
- SLAs must have the capacity to evolve, too!

Tooling matters.



@phrawzty

Obrigado!

DATADOG