honeycomb.io

# Pitfalls in Measuring SLOs

Danyel Fisher
@fisherdanyel

honeycomb.io

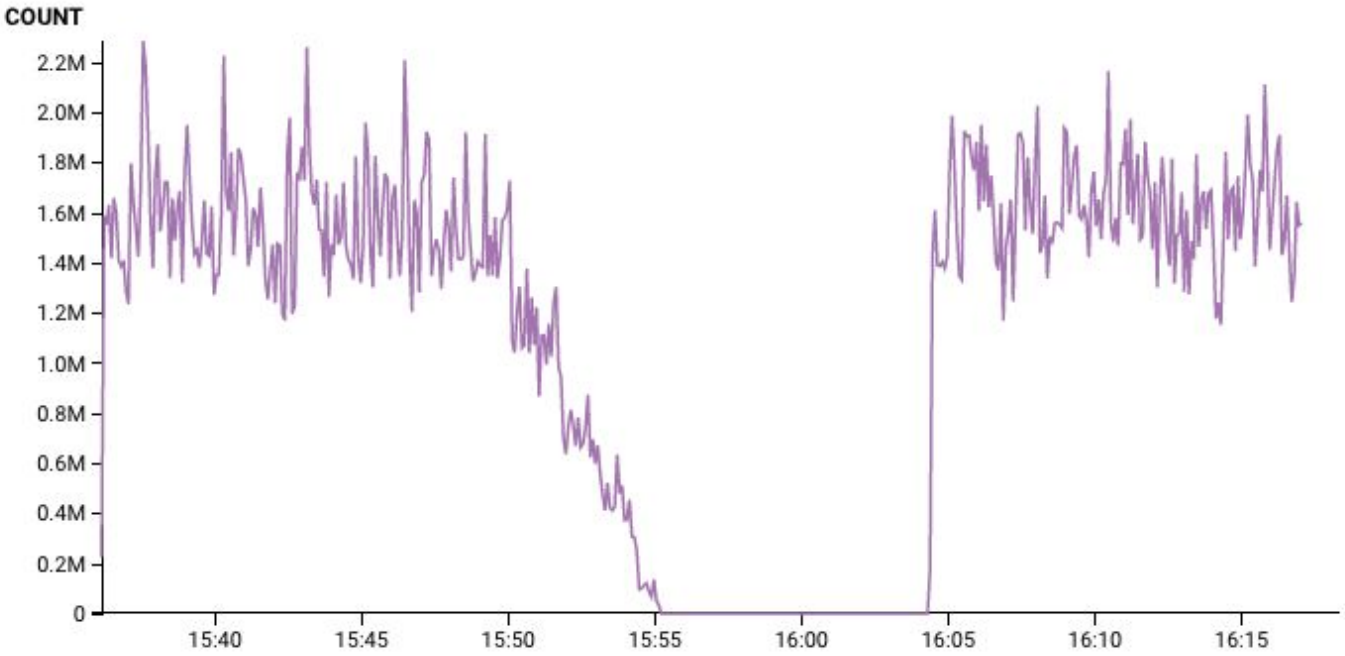
# An Outage

Danyel Fisher @fisherdanyel


Jul 9 2019, 12:00 AM – Jul 10 2019, 12:00 AM
Results
BubbleUp
Traces
Raw Data
Graph Settings
COUNT
45M
40M
35M
30M
25M
20M
15M
10M
5M
0
07/09 00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
Wed Jul 10

Jul 9 2019, 3:36 PM – Jul 9 2019, 4:17 PM

Results BubbleUp Traces Raw Data Graph Settings



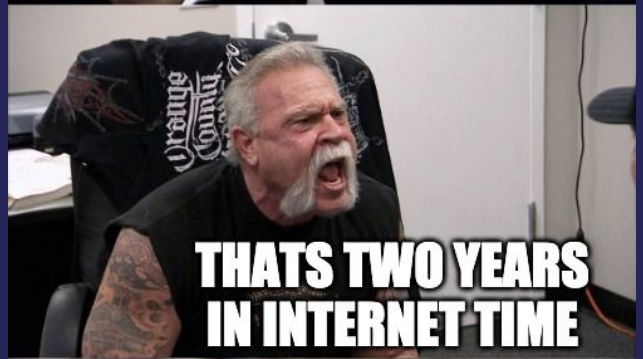
# How Broken is “Too Broken”?

What do you do when things break?

How bad was this break?


WE WERE DOWN FOR TWELVE MINUTES


TWELVE MINUTES
ISN'T VERY LONG


THATS TWO YEARS
IN INTERNET TIME


COMMERCIAL TRUST
Build new features!
LABOR TRUST
We need to improve quality!
STRIKE

# In search of a common language

Management

Engineering

Clients and Users

How broken is “too broken”?

What does “good enough” mean?

Combatting alert fatigue

# How do we do it?

A telemetry system **produces events** that correspond to **real world use**

We can describe some of these events as **eligible**

We can describe some of them as **good**

# SLI: Service Level Indicator

Given an **event**, is it **eligible? Is it good?**

**Eligible:** “Had an http status code”

**Good:** “... that was a 200, and was served under 500 ms”

# Defining Quality

# Defining Quality

$$\frac{\text{good events}}{\text{eligible events}}$$

## SLO

Minimum **Quality ratio** over a period of time

## Error Budget

Number of bad events allowed.

# Left over budget

Deploy faster

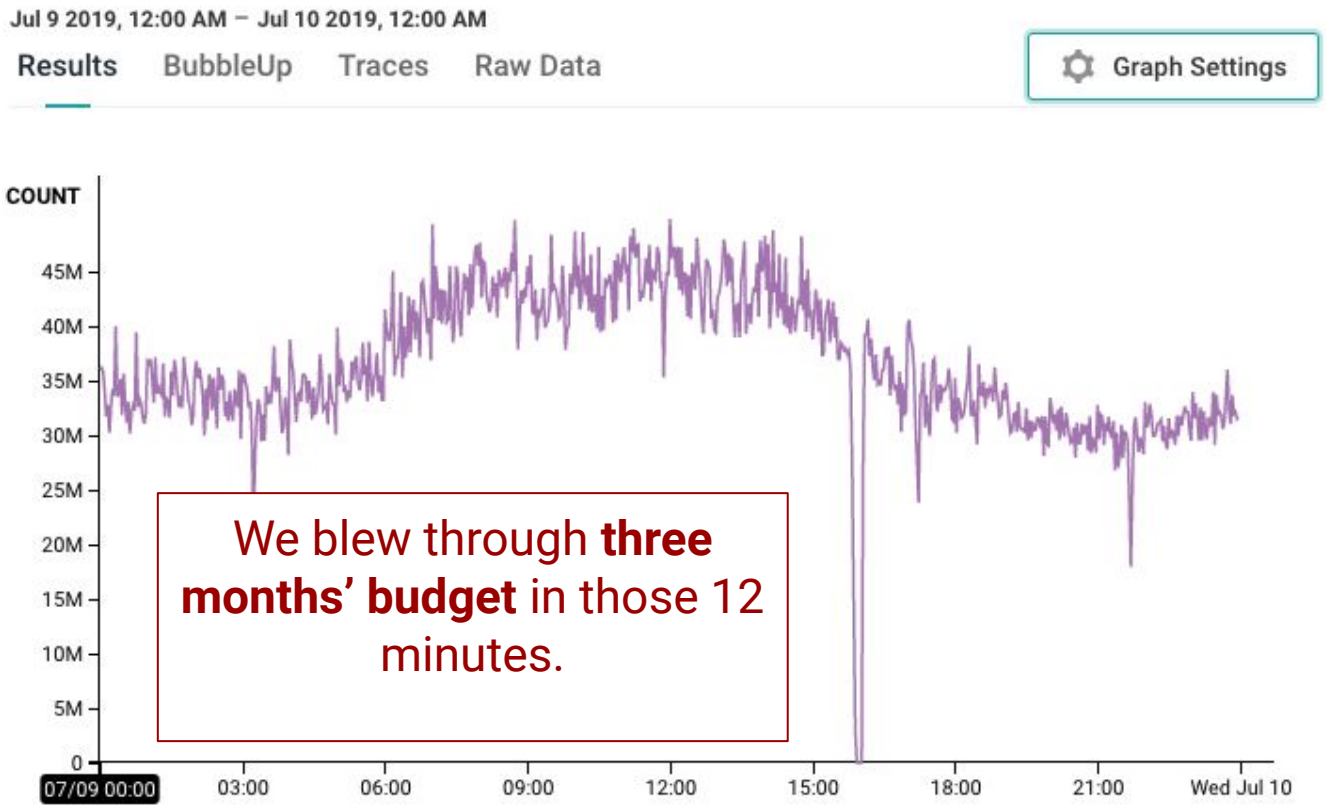
Room for experimentation

Opportunity to tighten SLO

# Honeycomb SLOs

| | | |
|---|---|---|
| We **always** store incoming user data | **99.99%** | **~4.3 minutes** |
| Default dashboards **usually** load in < 1s | **99.9%** | **45 minutes** |
| Queries **often** return in < 10 s | **99%** | **7.3 hours** |

# User Data Throughput




We blew through **three months' budget** in those 12 minutes.



honeycomb.io

# Ingest Outage

We dropped customer data

# Ingest Outage

We dropped customer data

We rolled it back (manually)

We communicated to customers

We halted deploys

# What happened?

We checked in code that **didn't build**.

We had **experimental** CI build wiring.

Our scripts deployed **empty binaries**.

There was **no health check** and rollback.

# How we fixed it

We **stopped writing new features**

We **prioritized stability**

We **mitigated risks**

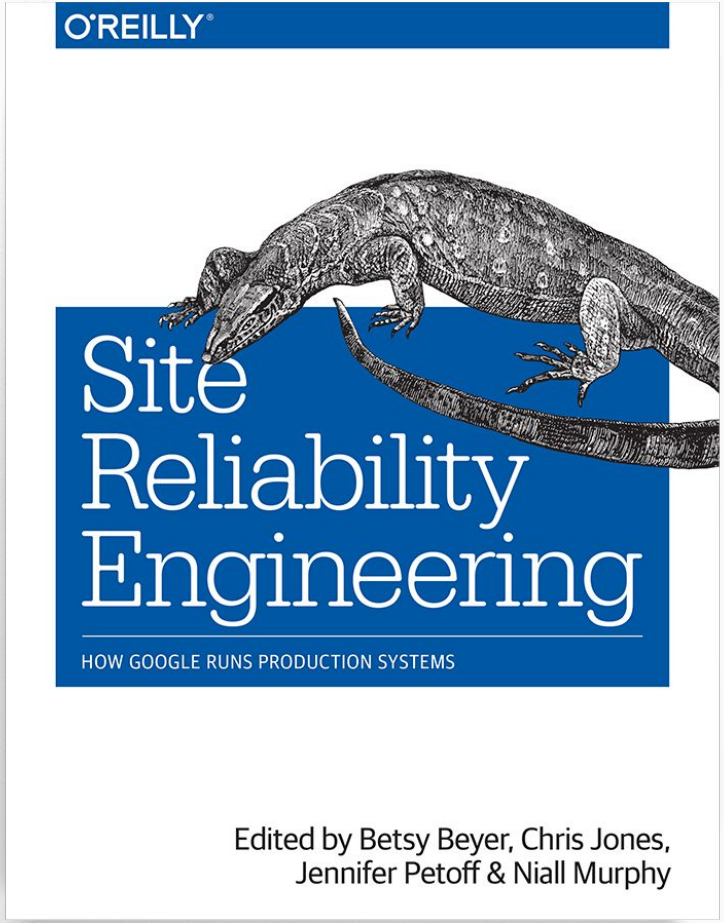
SLOs allowed us to characterize

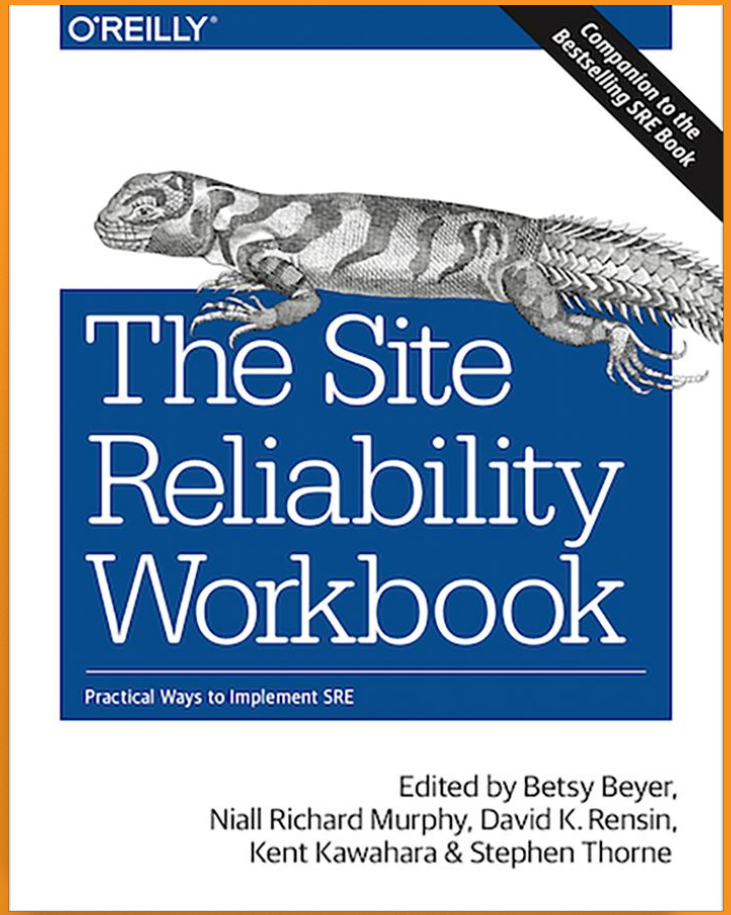
what went wrong,
how badly it went wrong,
and
how to prioritize repair

honeycomb.io

# Learning from SLOs

Danyel Fisher @fisherdanyel


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy


O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne

# This Talk

- Design Thinking
- Expressing and Viewing
- Burndown Alerts and Responding
- Learning from our Experiences
- Success Stories

# Design Thinking and Task Analysis

Understand user goals and needs

Learn from informants and experts

Collaborate with internal team

Collect feedback and ideas externally

# Displays and Views

# See where the burndown was happening, explain why, and remediate

# Expressing SLOs

**Event based**

“How many events had a duration < 500 ms”

**Time based**

“How many 5 minute periods, had a P95(duration) < 500 ms”

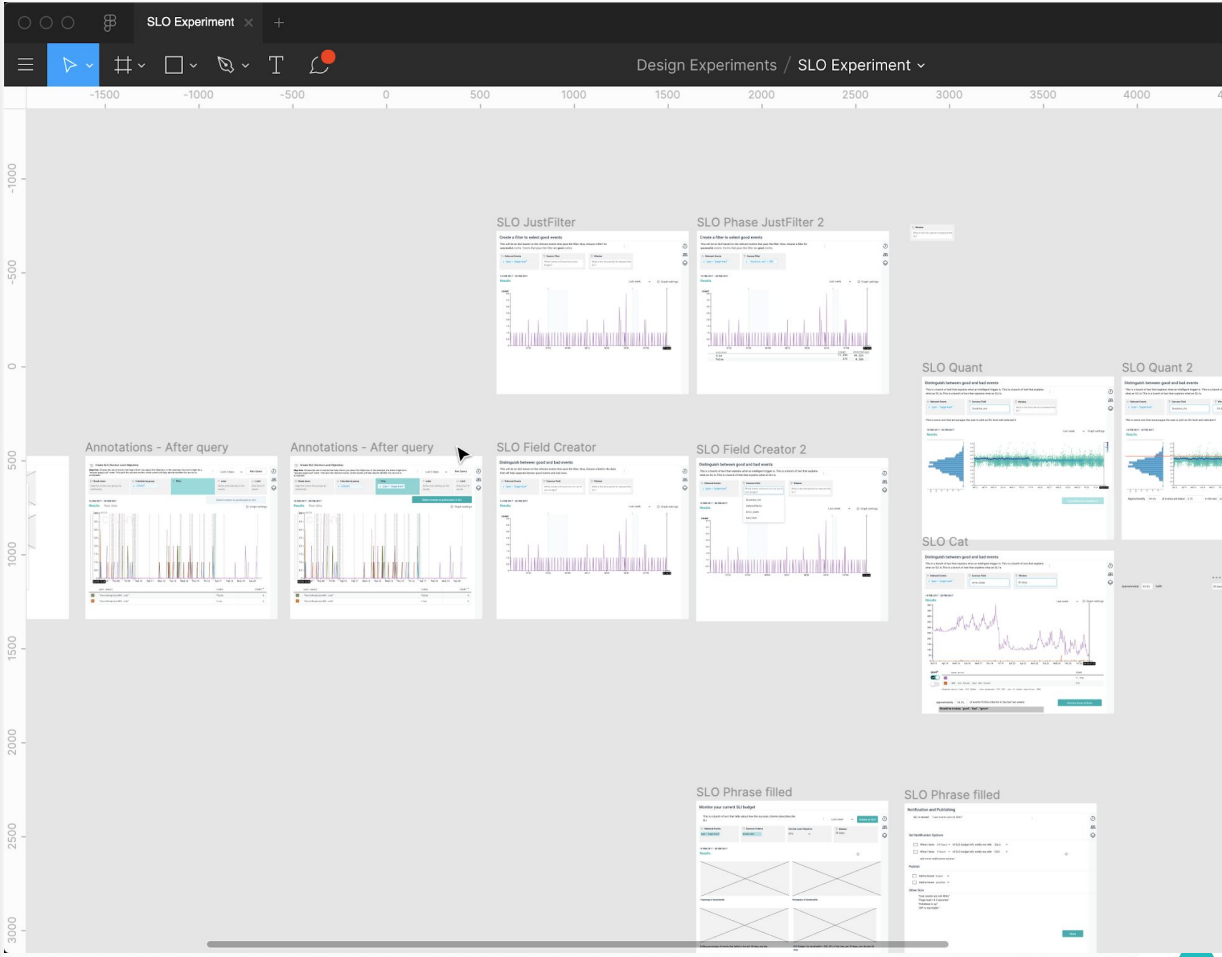
# How do we **express** SLOs?

Good events

Bad events

How often

Time range




Danyel Fisher @fisherdanyel

# How do we **express** SLOs?

Good events

Bad events

How often

Time range

## Distinguish between good and bad events

This is a bunch of text that explains what an intelligent trigger is. This is a bunch of text that explains what an SLI is.This is a bunch of text that explains what an SLI is.

**Relevant Events**: x type = "page-load"

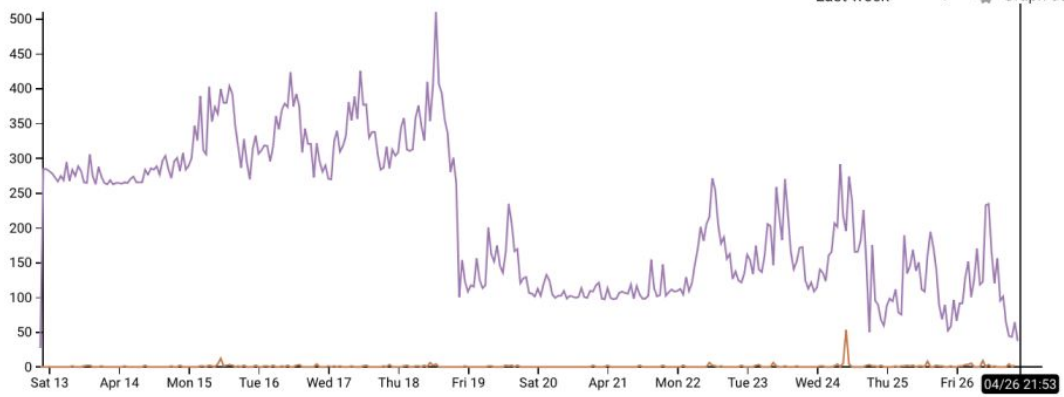
**Success Field**: error_state

**Window**: 30 days

15 FEB 2017 - 20 FEB 2017

Results

Last week | Graph settings




| good? | save_error | COUNT |
|---|---|---|
| ☑ | | 71,395 |
| ☐ | 404: not_found, User Not Found | 272 |

elapsed query time: 257.684ms rows examined: 247,382 pct of nodes reporting: 100%

Approximately 99.9% of events fit this criterion in the last two weeks

Choose these Criteria

Should be tristate: "good", "bad", "ignore"

Danyel Fisher @fisherdanyel

# How do we **express** SLOs?

Good events

Bad events

How often

Time range

**Eligible:** $name is “run_trigger_detailed”

**Good:** $app.error does not exist

```
IF(EQUALS($name, "run_trigger_detailed"), NOT(EXISTS($app.error)))
```

# How do we **express** SLOs?

Good events

Bad events

How often

Time range

```
IF(AND(NOT(CONTAINS($app.user.email, "@honeycomb.io")), NOT(EXISTS($app.user.sudoing)),
NOT(EQUALS($app.error, "Could not perform this action, this dataset has been
deleted")), NOT(EQUALS($response.status_code, 403)), EQUALS($handler.route,
"/{team_slug}/home/{slug}/query")), AND(LT($duration_ms, 250),
LT($response.status_code, 400)))
```

# How do we **express** SLOs?

Good events

Bad events

How often

Time range

## Update SLO

Learn more about creating SLOs

Name

Basset triggers query retriever successfully

Description

few errors in run_trigger_detailed

sli_errors field looks at app.error.

SLI Column

fx sli_errors

```
IF(EQUALS($name, "run_trigger_detailed"), NOT(EXISTS($app.error)))
```

Time Period (in days)

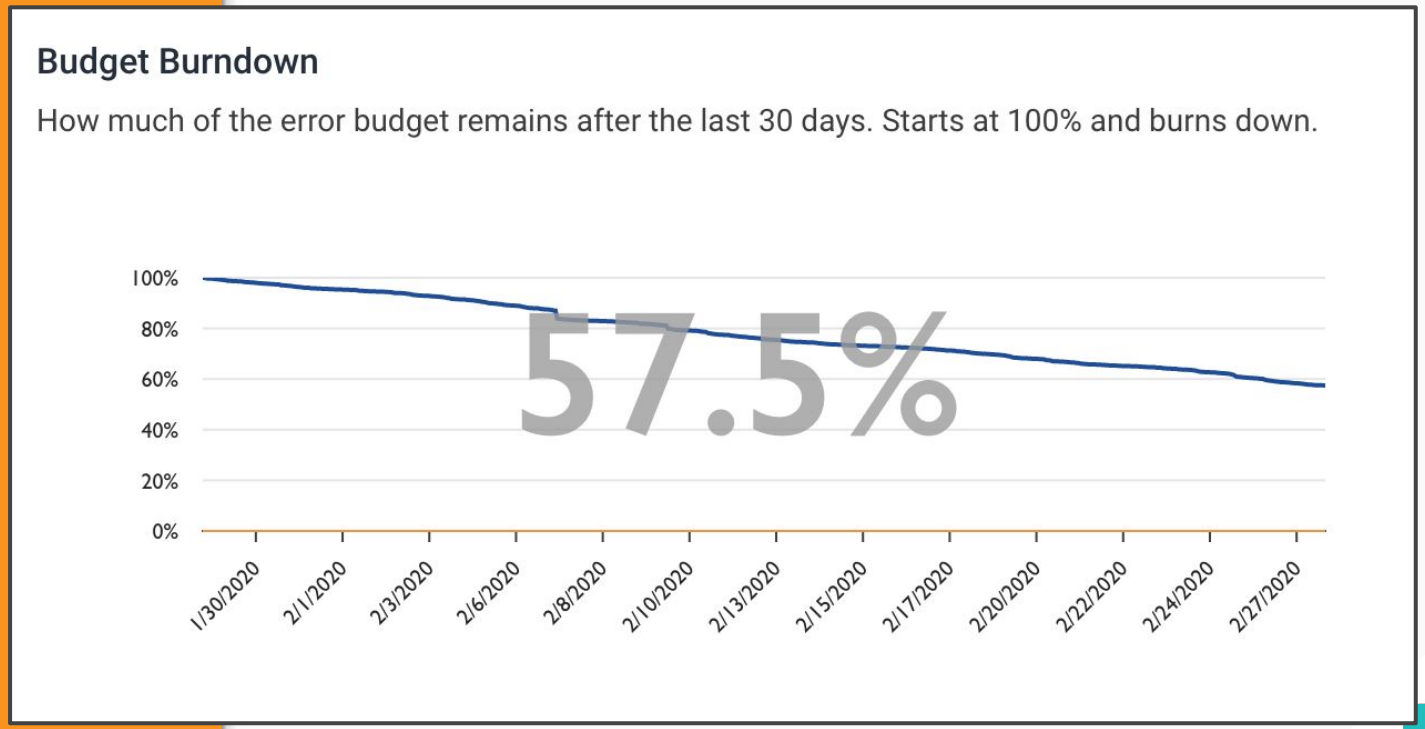
7

Target Percentage

99.99

Cancel | Update SLO

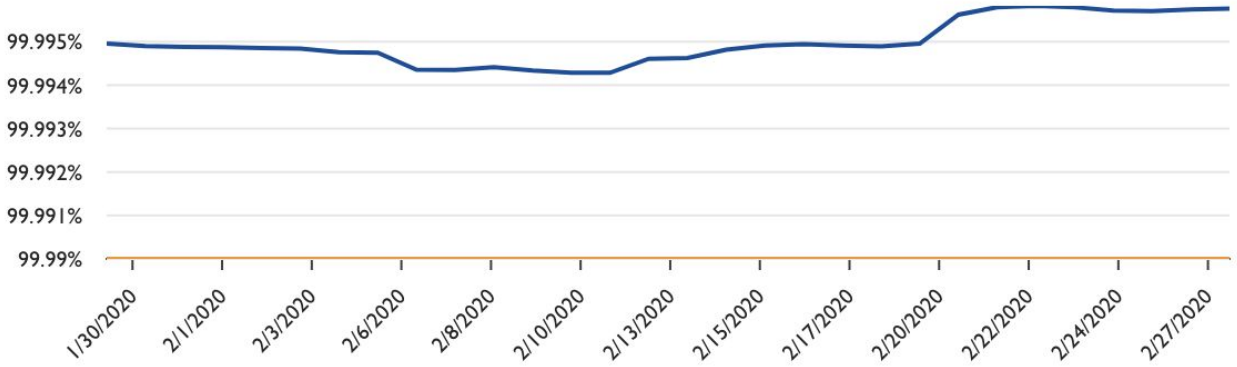
# Status of an SLO



# How have we done?

## Historical SLO Compliance

For each day of the past 30, how often this SLI has succeeded over the preceding 30 days.



# Rendering is fast enough

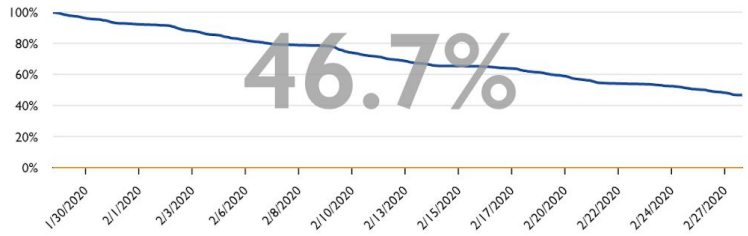
Edit SLO

95% of eligible events from the **user-events** column `page-load-sli` will succeed over a period of 30 days.

| Exhaustion time | Status |
|---|---|
| 83 hours | Normal |

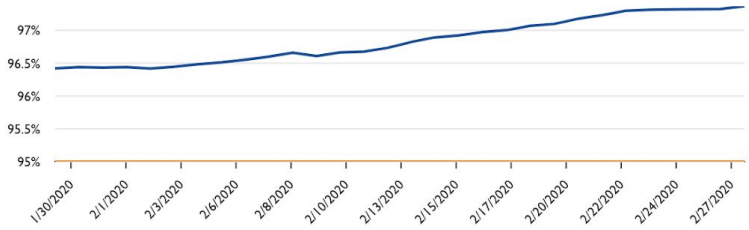
## Budget Burndown

How much of the error budget remains after the last 30 days. Starts at 100% and burns down.




## Historical SLO Compliance

For each day of the past 30, how often this SLI has succeeded over the preceding 30 days.



# Poodle Home Page

Edit SLO

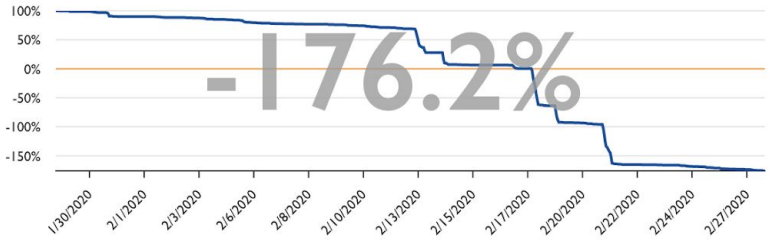
/home loads in <250 ms and doesn't return an error

99.5% of eligible events from the **poodle** column `sli_on_home_page_load_noerr` will succeed over a period of 30 days.

| Exhaustion time | Status |
|---|---|
| 0 hours | Triggered |
| 4 hours | Triggered |

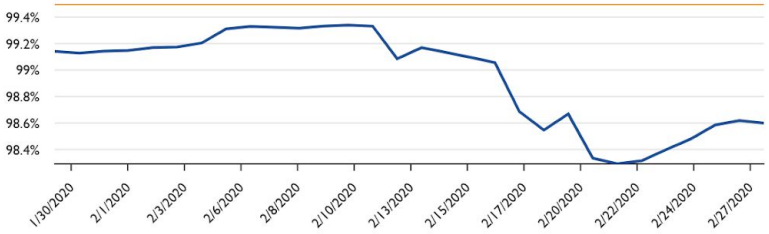
## Budget Burndown

How much of the error budget remains after the last 30 days. Starts at 100% and burns down.




## Historical SLO Compliance

For each day of the past 30, how often this SLI has succeeded over the preceding 30 days.



# Latency per-event

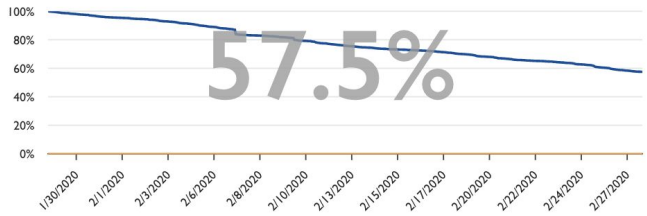
Edit SLO

99.99% of eligible events from the shepherd column sli
will succeed over a period of 30 days.

| Exhaustion time | Status |
|---|---|
| 4 hours | Normal |

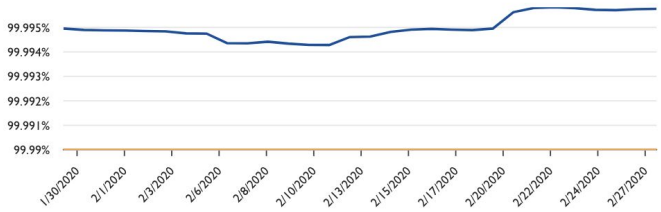
## Budget Burndown

How much of the error budget remains after the last 30 days. Starts at 100% and burns down.




## Historical SLO Compliance

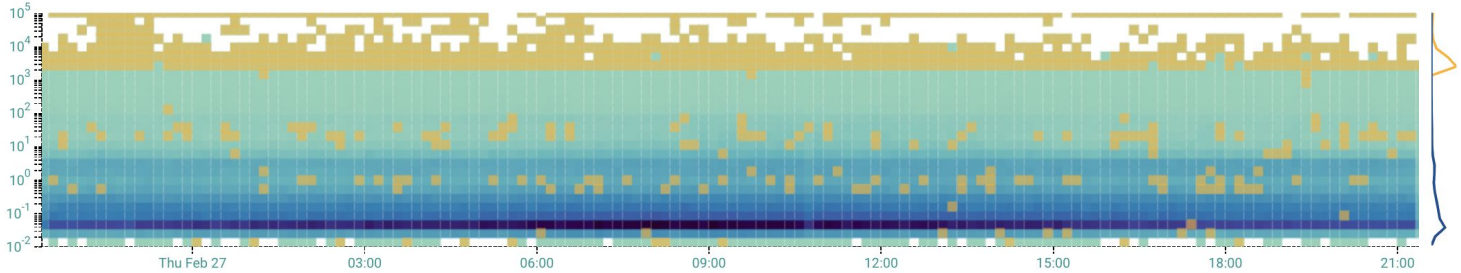
For each day of the past 30, how often this SLI has succeeded over the preceding 30 days.




## Distribution of Events failing SLI by duration_ms

Last 24 hours

**Feb 26 2020, 9:23 PM — Feb 27 2020, 9:23 PM**



# Poodle Home Page

Edit SLO

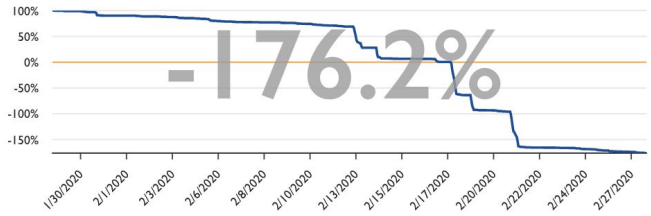
/home loads in <250 ms and doesn't return an error

99.5% of eligible events from the poodle column `sli_on_home_page_load_noerr` will succeed over a period of 30 days.

| Exhaustion time | Status |
|---|---|
| 0 hours | Triggered |
| 4 hours | Triggered |

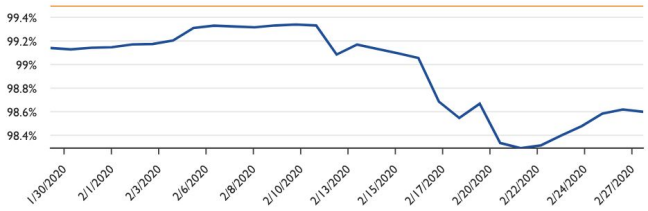
## Budget Burndown

How much of the error budget remains after the last 30 days. Starts at 100% and burns down.




## Historical SLO Compliance

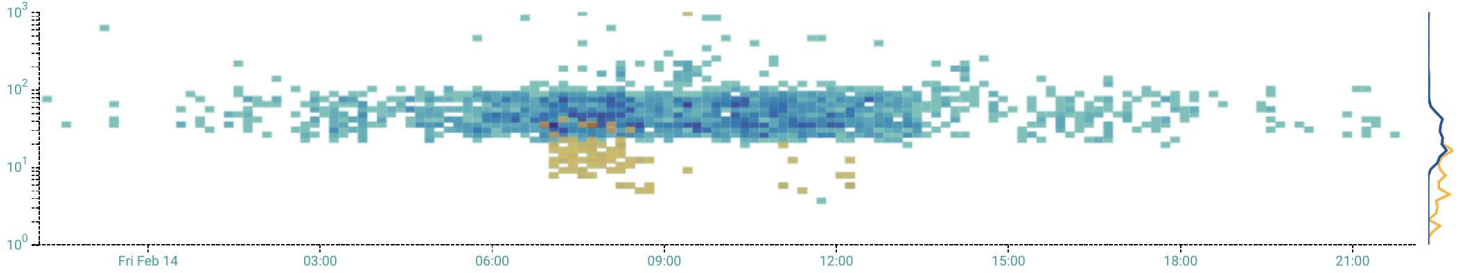
For each day of the past 30, how often this SLI has succeeded over the preceding 30 days.




## Distribution of Events failing SLI by `duration_ms`

14 days ago to 13 days ago

**Feb 13 2020, 10:06 PM — Feb 14 2020, 10:06 PM**



# What went wrong?

High dimensional data

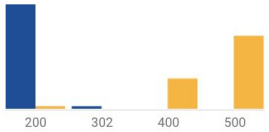
High cardinality data

## Dimensions

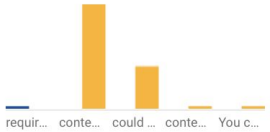
Try to GROUP BY columns that look most different between the successful and failed.

1-15 of 54

flt response.stat…




str app.error




flt rollup.db.cal…


str request.heade…

str request.heade…

str app.user.email

str app.dataset.n…

str app.dataset.s…

str app.team.name

str app.team.slug

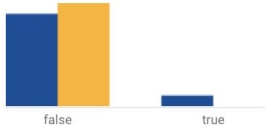
str gorilla.vars.…

str gorilla.vars.…

str request.path

str request.url

boo app.flag.disa…




## Measures

Try asking WHERE the successful and failed are most different.

1-5 of 8

flt duration_ms

flt rollup.db.dur…

flt app.dataset.id

flt app.user.id

flt global.proces…

## Dimensions

Try to GROUP BY columns that look most different between the successful and failed.

1-15 of 54

flt response.stat…

200 302 400 500

str app.error

requir… conte… could … conte… You c…

flt rollup.db.cal…

2 4 6 7 8 9 10 13 14 15 16 17 18 19 20

str request.heade…

str request.heade…

str app.user.email

str app.dataset.n…

str app.dataset.s…

str app.team.name

str app.team.slug

str gorilla.vars.…

str gorilla.vars.…

str request.path

str request.url

boo app.flag.disa…

false true

## Measures

Try asking WHERE the successful and failed are most different.

1-5 of 8

flt duration_ms

flt rollup.db.dur…

flt app.dataset.id

flt app.user.id

flt global.proces…

## Dimensions

Try to GROUP BY columns that look most different between the successful and failed.

1-15 of 54

flt response.stat…

200 302 400 500

str app.error

requir… conte… could …

flt rollup.db.

Value
could not find team
Baseline 0%
Selection 28%

Click column for actions

str app.dataset.n…

str app.datase

str request.heade…

str request.heade…

str app.user.email

str app.team.name

str app.team.slug

str gorilla.vars.…

str gorilla.vars.…

str request.path

str request.url

boo app.flag.disa…

false true

## Measures

Try asking WHERE the successful and failed are most different.

1-5 of 8

flt duration_ms

flt rollup.db.dur…

flt app.dataset.id

flt app.user.id

flt global.proces…

# See where the burndown was happening, explain why, and remediate

# User Feedback

"The Bubble Up in the SLO page is really powerful at highlighting **what is contributing the most** to missing our SLIs, it has definitely confirmed our assumptions."

# User Feedback

"Your customers have to be happy... we have to have an understanding of the customer experience. … **To the millisecond we knew what our percentage was of success versus failure**."

-Josh Hull, Site Reliability Engineering Lead, Clover Health

# User Feedback

“The historical SLO chart also **confirms a fix** for a performance issue we did greatly contributed to the SLO compliance by showing a nice upward trend line. :)”

# User Feedback

"I'd love to drive alerts off our SLOs. Right now **we don't have anything to draw us in** and have some alerts on the average error rate but they're a little spiky to be useful. It would be great to get a better sense of when the budget is going and define alerts that way."

# Burndown Alerts

# How is my system doing?

Am I over budget?

When will my alarm fail?

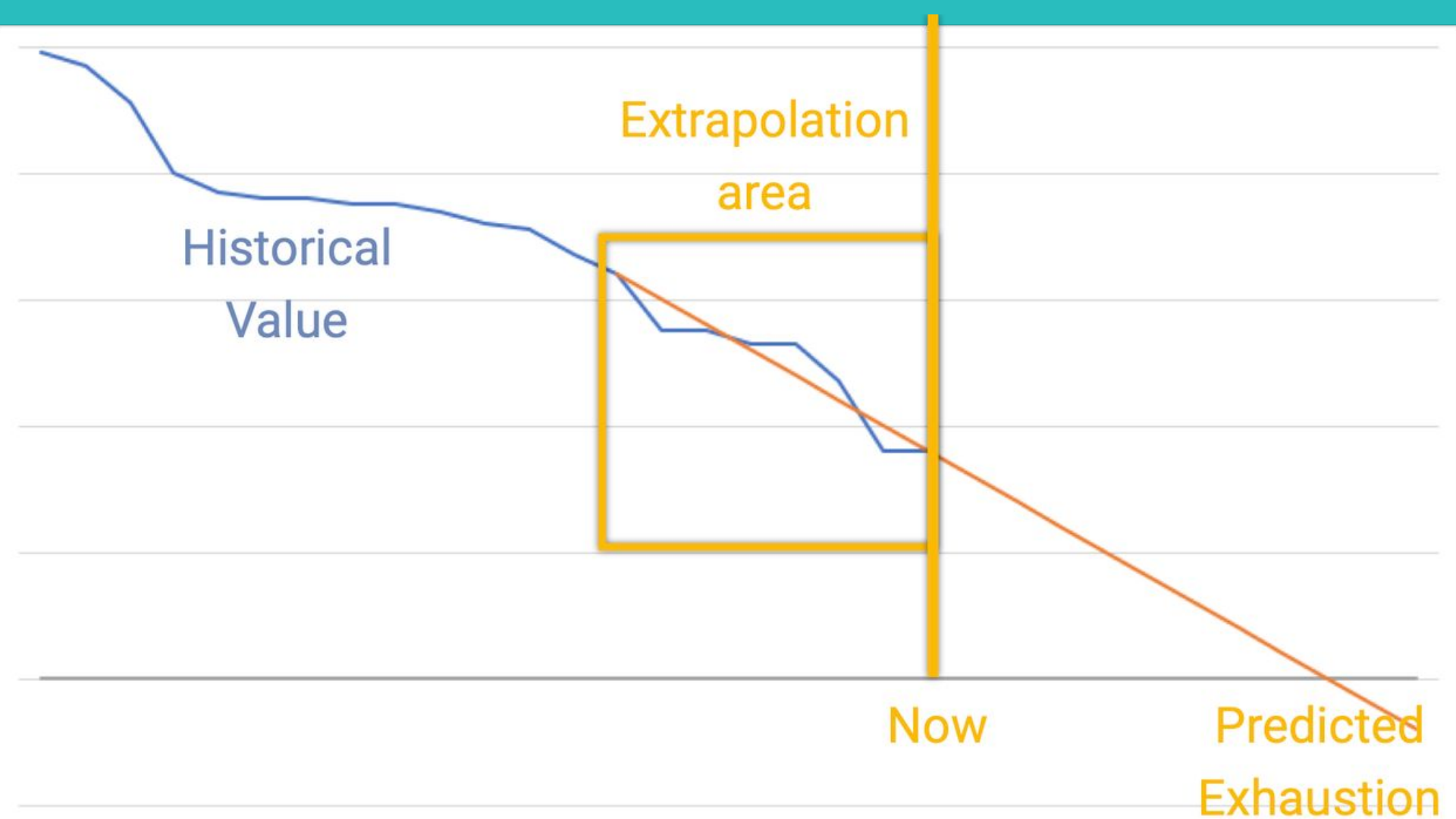
# When will I fail?

**User goal: get alerts to exhaustion time**

Human-digestible units

- 24 hours: “I’ll take a look in the morning”
- 4 hours: “All hands on deck!”


Historical
Value
Now


Extrapolation
area
Historical
Value
Now
Predicted
Exhaustion

# Implementing Burn Alerts

Run a 30 day query

at a 5 minute resolution

every minute

Datasets / AWS Cost and Usage

Add a description for this query

| VISUALIZE | WHERE | GROUP BY | ORDER BY | LIMIT |
|---|---|---|---|---|
| SUM(amortizedCost) | None; include all rows | resourceTags/user:Role<br>product/productFamily | SUM(amortizedCost) desc | None |

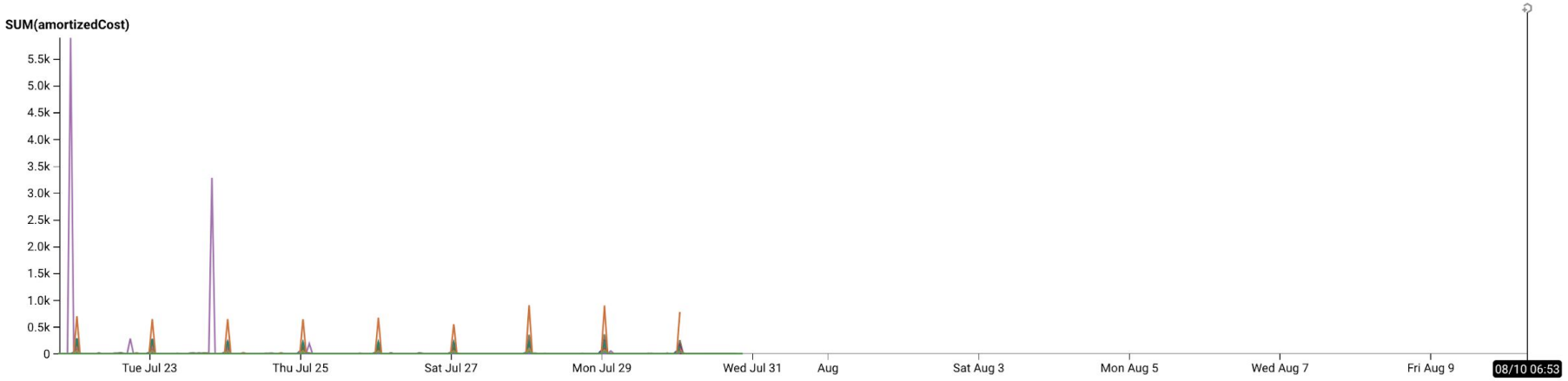
Run Query

Run 4 months ago

Jul 21 2019, 7:34 PM – Aug 10 2019, 6:53 AM

Results BubbleUp Traces Raw Data

Graph Settings




| resourceTags/user:Role | product/productFamily | SUM(amortizedCost) |
|---|---|---|
| | | 9,744.61691 |
| kafka | Data Transfer | 6,492.72813 |
| retriever | Compute Instance | 2,374.18905 |
| kafka | Storage | 1,991.24955 |

# Caching is Fun!

# Fun with Caching

Vital to cache results

… but not incomplete results

… … at what resolution of cache?

# Flappy Alerts

“It’ll expire at 3:55”

“Wait, make that 4:05”

“Nope, 3:55 again!”

**(We added a 10%ish buffer)**

# Recovering from Bankruptcy

A failure a month ago brought us to -169% and still hasn't aged out?

That means we don't get alerts anymore

Customer workaround: delete and re-create the SLO, thus blowing the cache

# Learning from Experience

Danyel Fisher @fisherdanyel

# Volume is important

Tolerate at least dozens of bad events per day

# Faults

```
err = zReader.Reset(bodyReader)
if err != nil {
    beeline.AddField(ctx, "dropped", "our fault")
    beeline.AddField(ctx, "drop_reason", err.Error())
    return 0, err
}
```

```
info, partitions, err := a.getReqInfo(r, reqinfo.ResourceEvents, a.prepPartitionFn(ctx))
if err != nil {
    if errors.Is(err, reqinfo.ErrDisabledWriteKey) {
        beeline.AddField(ctx, "dropped", "their fault")
        beeline.AddField(ctx, "drop_reason", "disabled write key")
        return apierr.MsgUnknownTeam
    }
    return err
}
```

# SLOs for Customer Service


Danyel Fisher @fisherdanyel

# Blackouts are easy

… but brownouts are much more interesting

1:29 AM **Honeycomb Dogfood** APP Honeycomb SLOs

**Triggered:** `Shepherd ALB timeout/error` **will violate SLO in 4h0m0s**

**Status**

The error budget for SLO `Shepherd ALB timeout/error` is in danger of being exhausted.

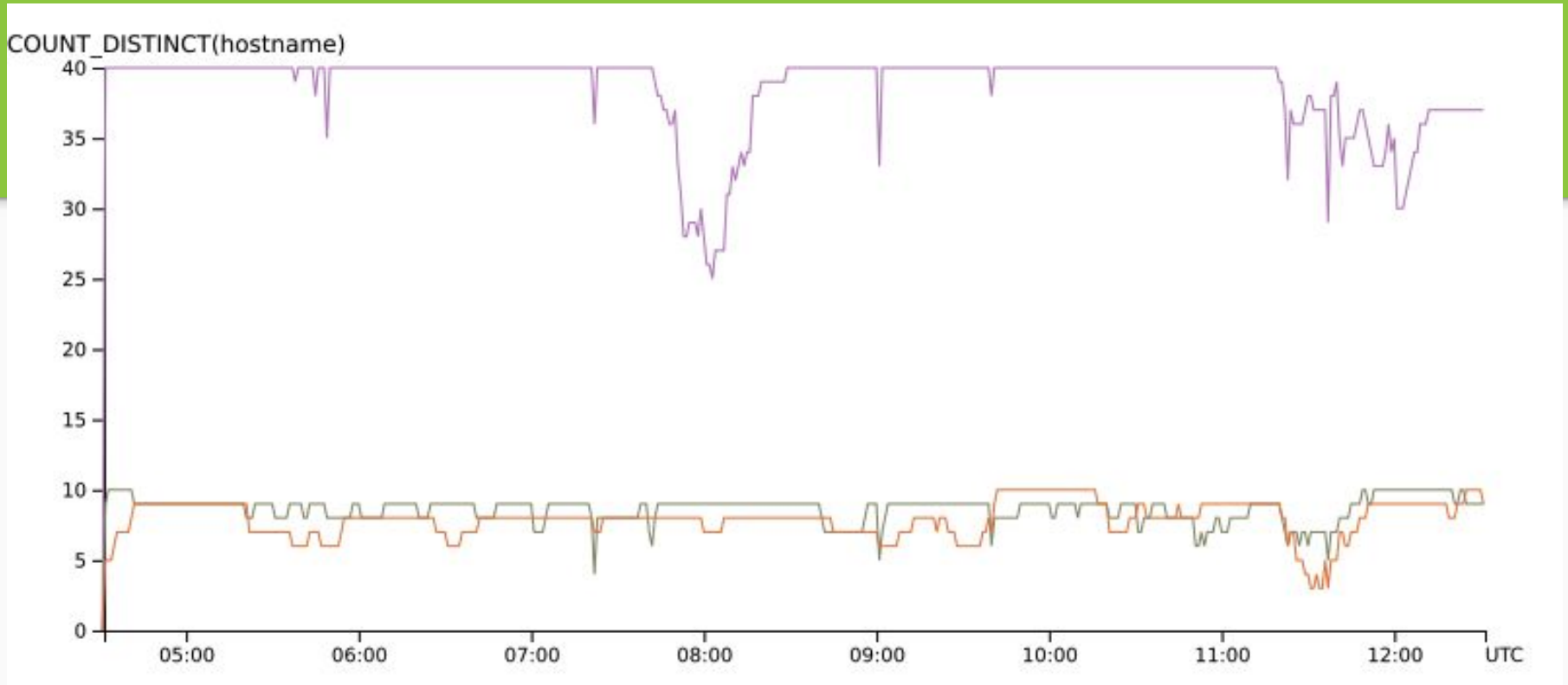
Description: `99.995% of ALB response codes should match what Shepherd gave us.

If these differ, it's either due to client misbehavior or due to Shepherd failing to respond.`

View SLO

# Timeline

| | |
|---|---|
| 1:29 am | SLO alerts. “Maybe it’s just a blip”<br>1.5% brownout for 20 minutes |


COUNT_DISTINCT(hostname)
40
35
30
25
20
15
10
5
0
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
UTC

# Timeline

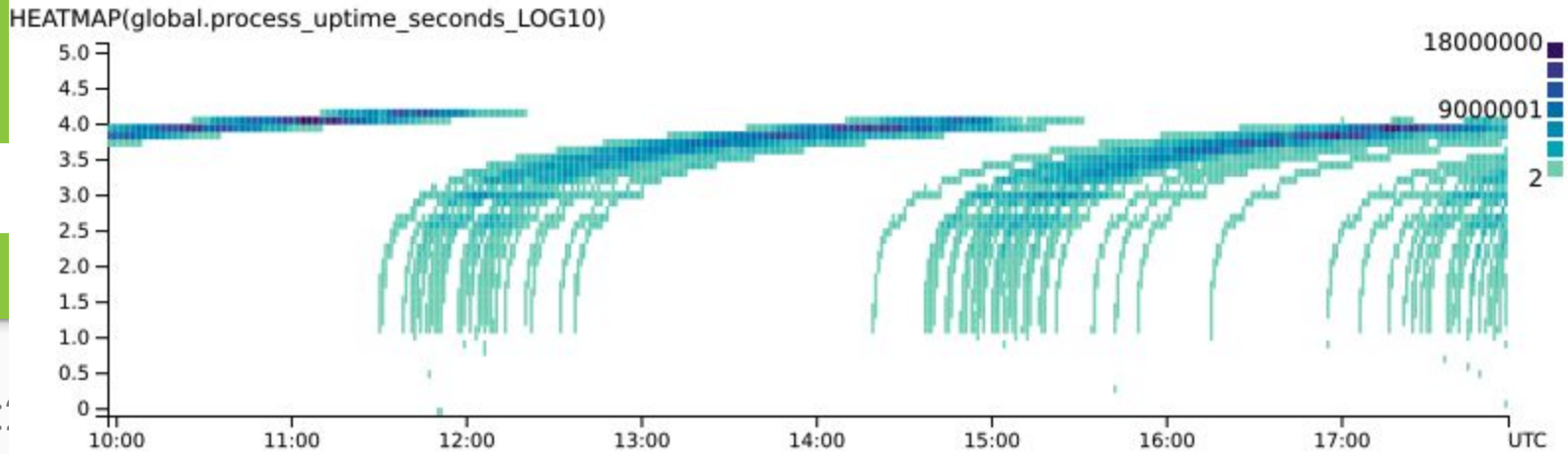
| | |
|---|---|
| 1:29 am | SLO alerts. “Maybe it’s just a blip” |
| | 1.5% brownout for 20 minutes |
| 4:21 am | Minor incident. “It might be an AWS problem” |

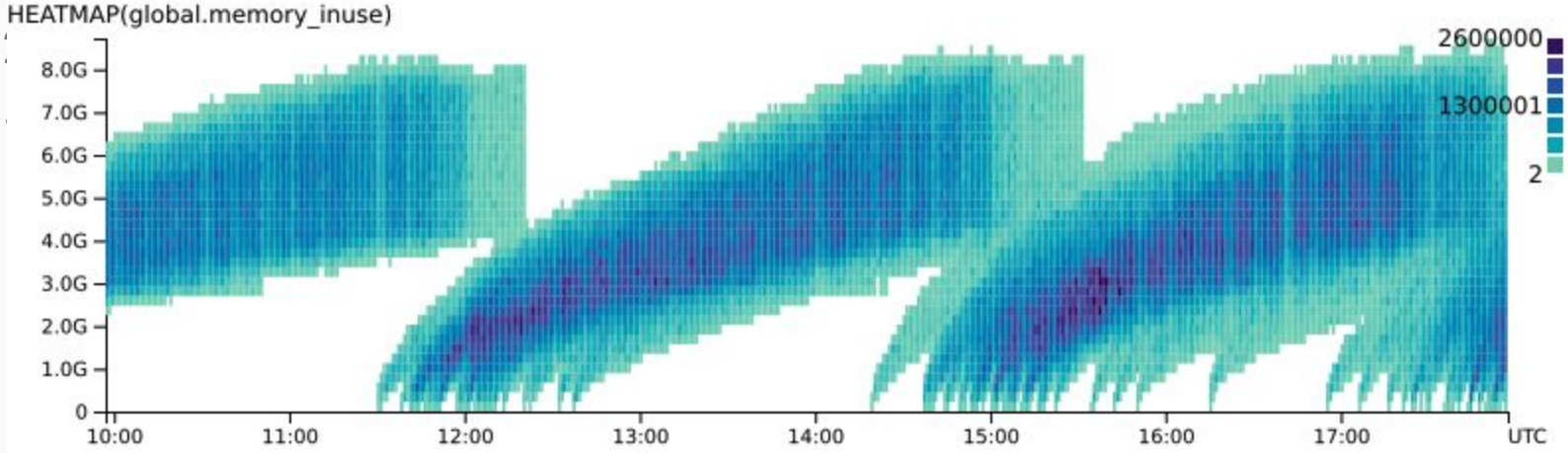
# Timeline

| | |
|---|---|
| 1:29 am | SLO alerts. “Maybe it’s just a blip” |
| | 1.5% brownout for 20 minutes |
| 4:21 am | Minor incident. “It might be an AWS problem” |
| 6:25 am | SLO alerts again. “Could it be ALB compat?” |

# Timeline

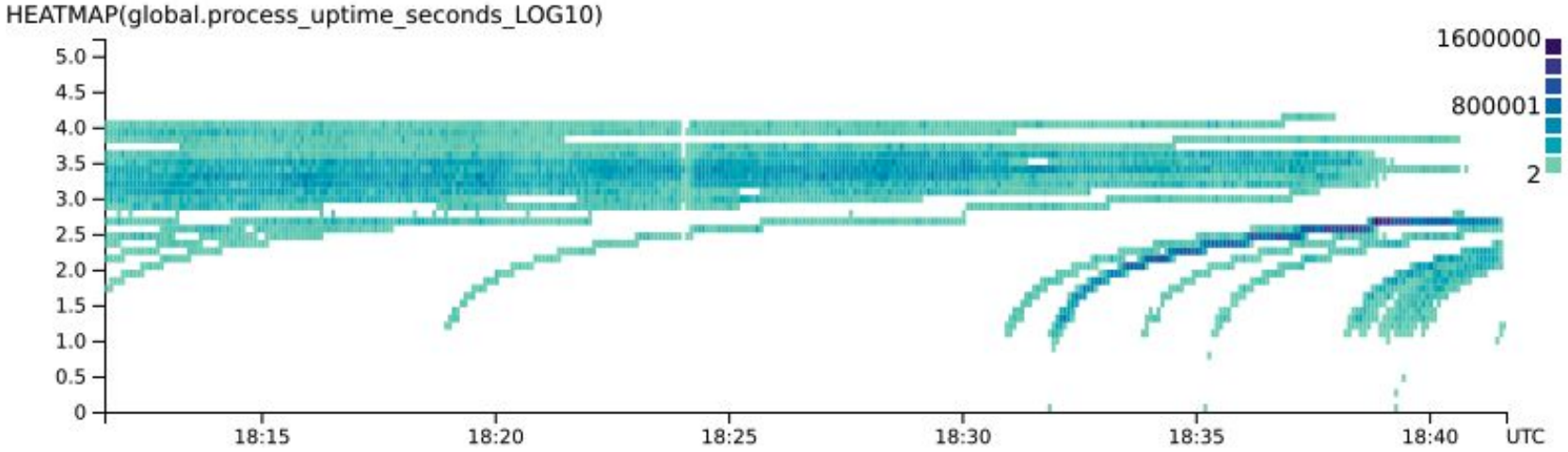
| | |
|---|---|
| 1:29 am | SLO alerts. “Maybe it’s just a blip” |
| | 1.5% brownout for 20 minutes |
| 4:21 am | Minor incident. “It might be an AWS problem” |
| 6:25 am | SLO alerts again. “Could it be ALB compat?” |
| 9:55 am | “Why is our system uptime dropping to zero?” |
| | It’s out of memory |
| | We aren’t alerting on that crash |

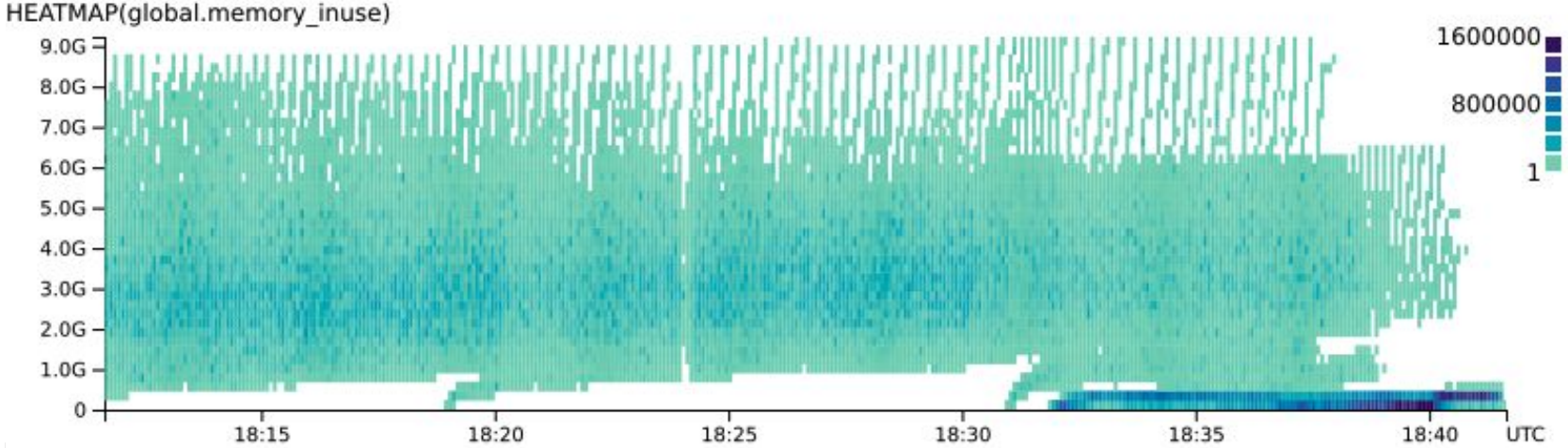

HEATMAP(global.process_uptime_seconds_LOG10)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
UTC
18000000
9000001
2


HEATMAP(global.memory_inuse)
8.0G
7.0G
6.0G
5.0G
4.0G
3.0G
2.0G
1.0G
0
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
UTC
2600000
1300001
2

# Timeline

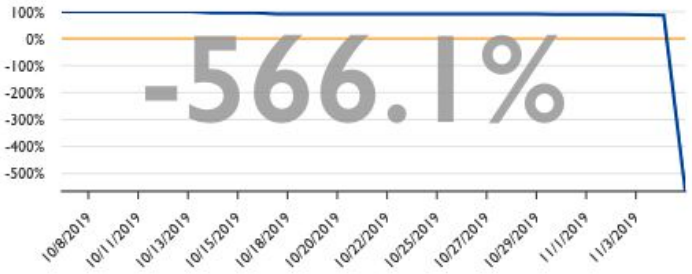
| | |
|---|---|
| 1:29 am | SLO alerts. “Maybe it’s just a blip” |
| | 1.5% brownout for 20 minutes |
| 4:21 am | Minor incident. “It might be an AWS problem” |
| 6:25 am | SLO alerts again. “Could it be ALB compat?” |
| 9:55 am | “Why is our system uptime dropping to zero?” |
| | It’s out of memory |
| | We aren’t alerting on that crash |
| 10:32 am | Fixed |


HEATMAP(global.process_uptime_seconds_LOG10)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
18:15
18:20
18:25
18:30
18:35
18:40
UTC
1600000
800001
2


HEATMAP(global.memory_inuse)
9.0G
8.0G
7.0G
6.0G
5.0G
4.0G
3.0G
2.0G
1.0G
0
18:15
18:20
18:25
18:30
18:35
18:40
UTC
1600000
800000
1

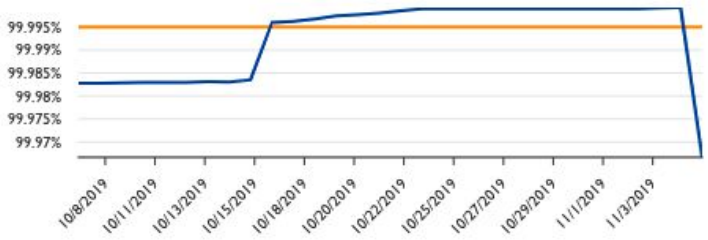
## Remaining Budget

How much of the error budget remains after the last 30 days.




## Historical SLO Compliance

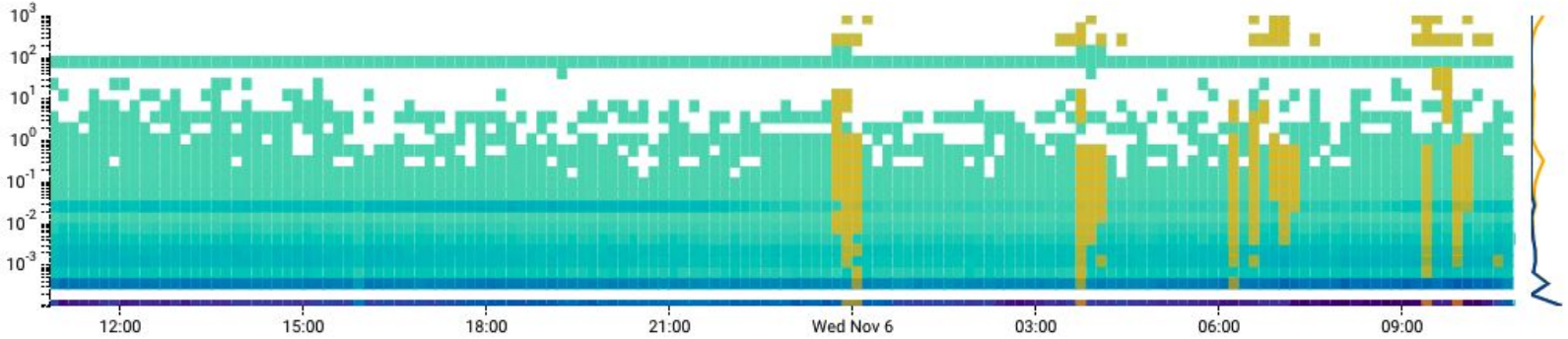
For each day of the past 30, how often this SLO has succeeded over the preceding 30 days.




## Distribution of Events failing SLI by flt backend_processing_time

Last 24 hours

Nov 5 2019, 10:52 AM – Nov 6 2019, 10:52 AM




Danyel Fisher @fisherdanyel

# How we fixed it

We **stopped writing new features**

We **prioritized stability**

We **mitigated risks**

**… and we promoted our SLO burn alerts**

# Cultural Change

It’s hard to replace alerts with SLOs

But a clear incident can help

# Reduce Alarm Fatigue

Focus on **user-affecting** SLOs

Focus on **actionable** alarms

honeycomb.io

# Conclusion

SLOs allowed us to characterize

- what went wrong,
- how badly it went wrong,
  - and
- how to prioritize repair

You can do it too
And maybe avoid our mistakes

# Pitfalls in Measuring SLOs

**Email:** danyel@honeycomb.io

**Twitter:** @fisherdanyel

Visit our booth on the 5th floor to kick the SLO tires, and to learn how we debug in high-res

hny.co/danyel