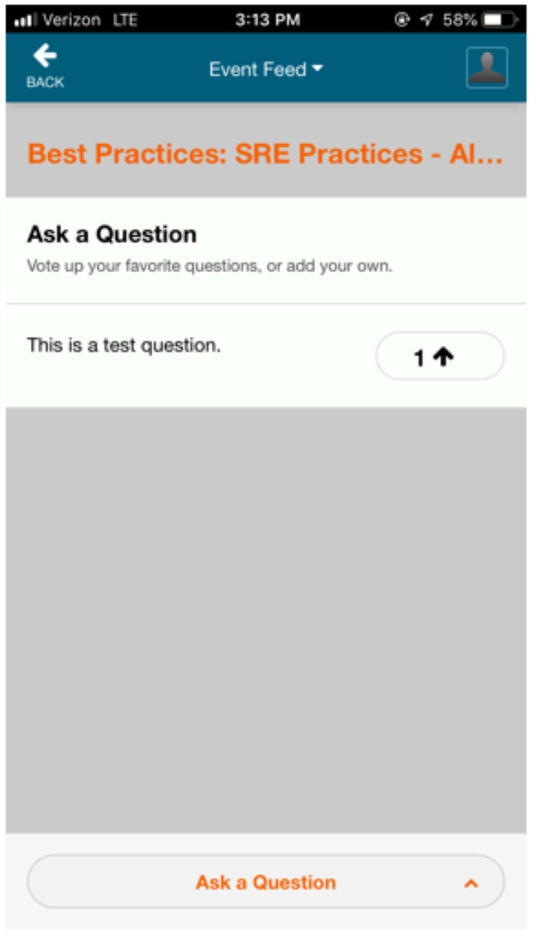
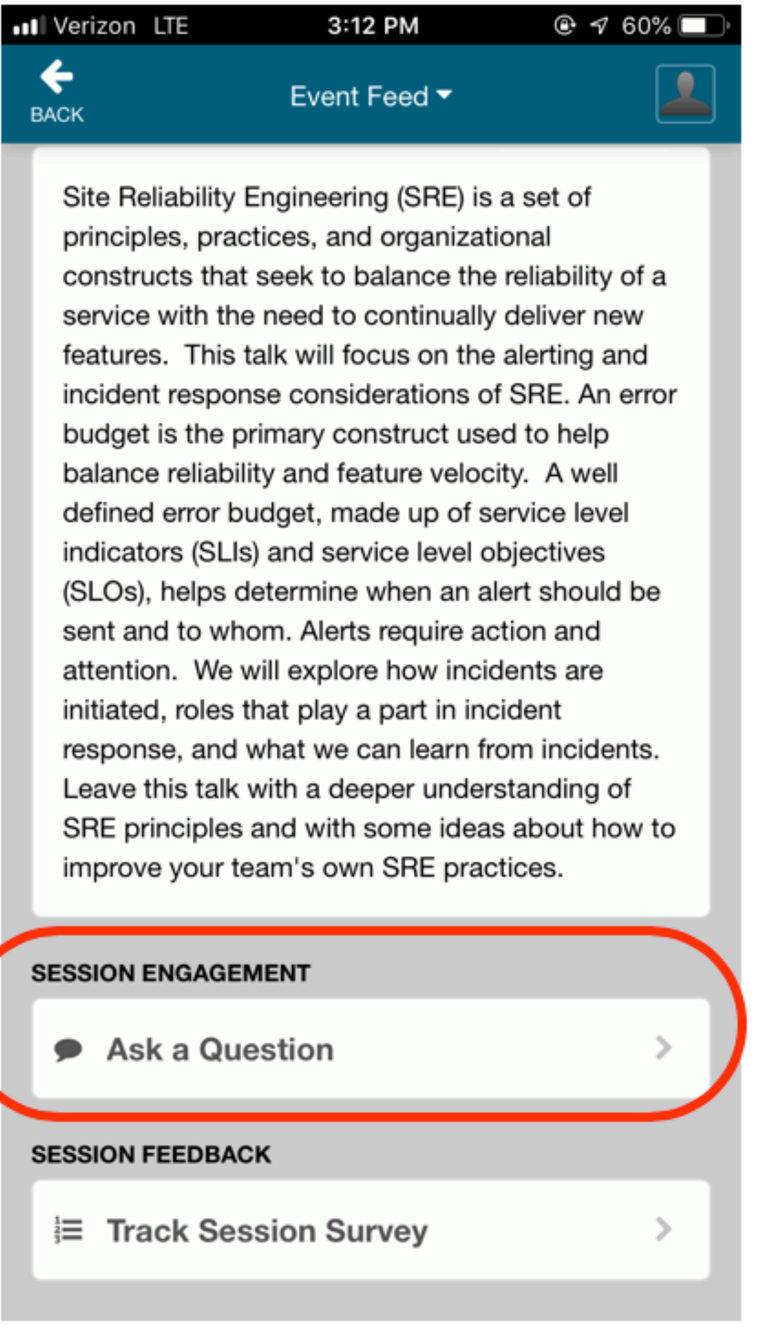
Submit your questions





Ask and upvote questions via the PagerDuty mobile app!
We will do Q&A at the end of the session.

Find Session > Session Engagement

getting comfortable in prod

to improve your life in dev

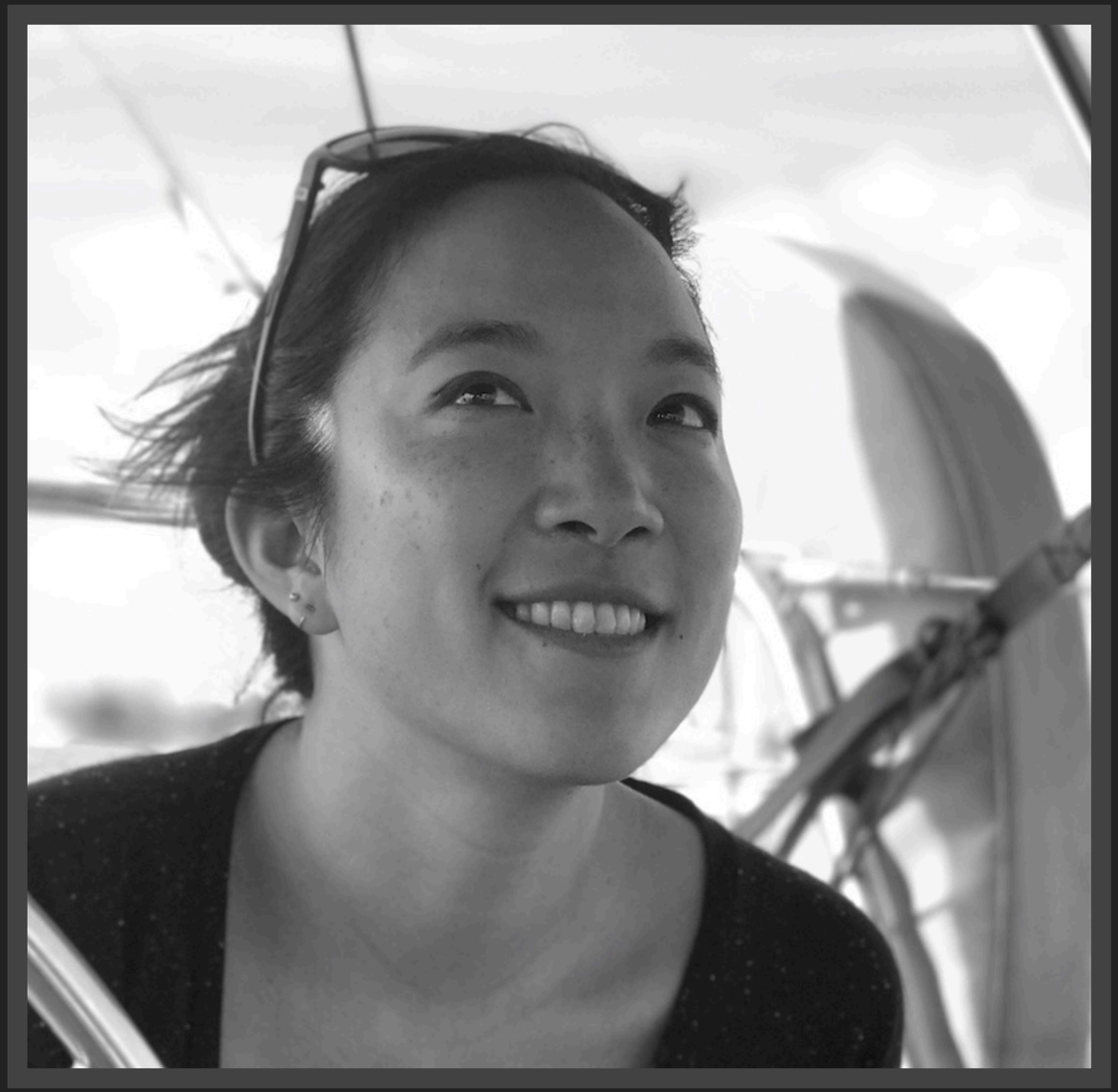
@cyen

@honeycombio

first, some background…

Christine

DEV



DEV

WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT

DEV OPS

WRITE → TEST → COMMIT → RELEASE

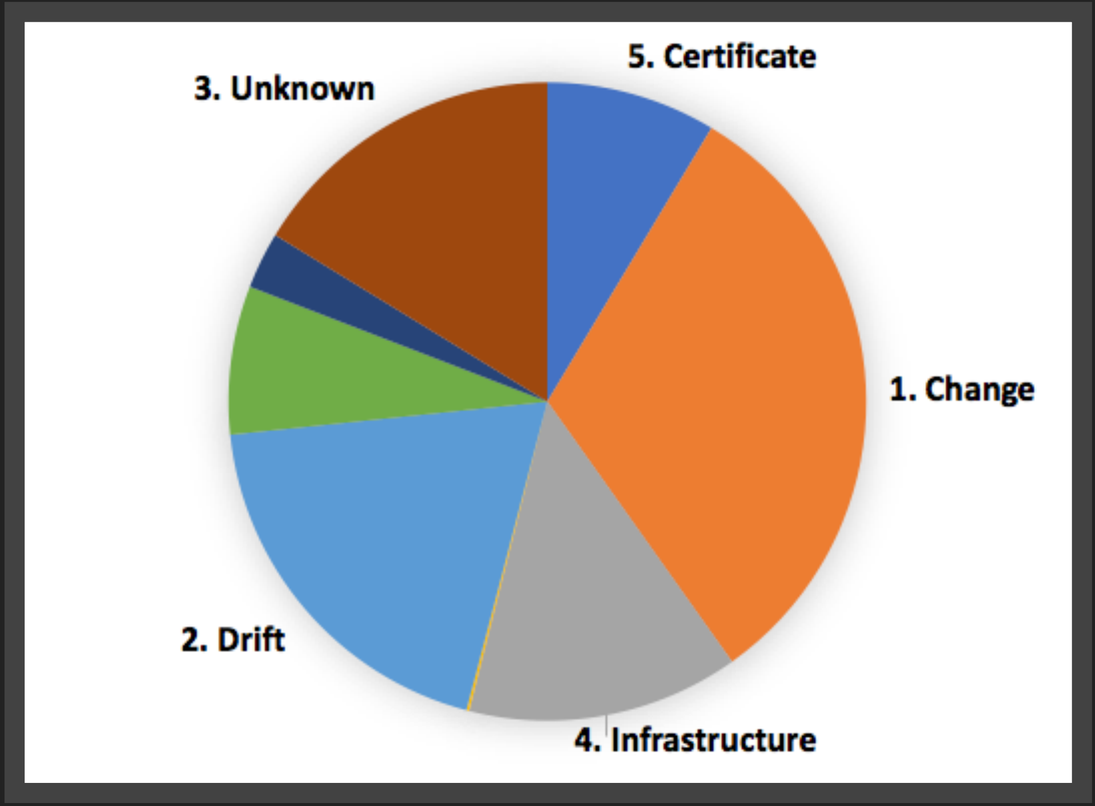
💥 → DEBUG → FIX

DEV 💥 OPS

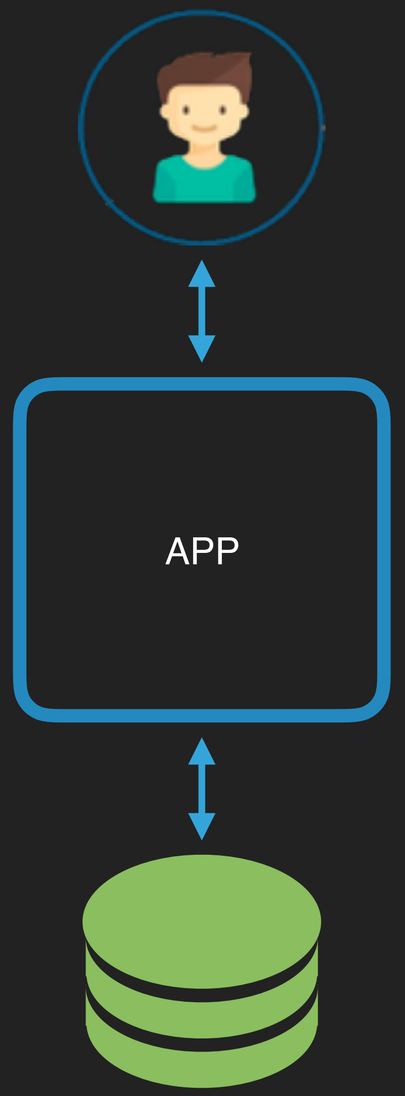
"Works on my machine"

"The only good diff is a red diff"

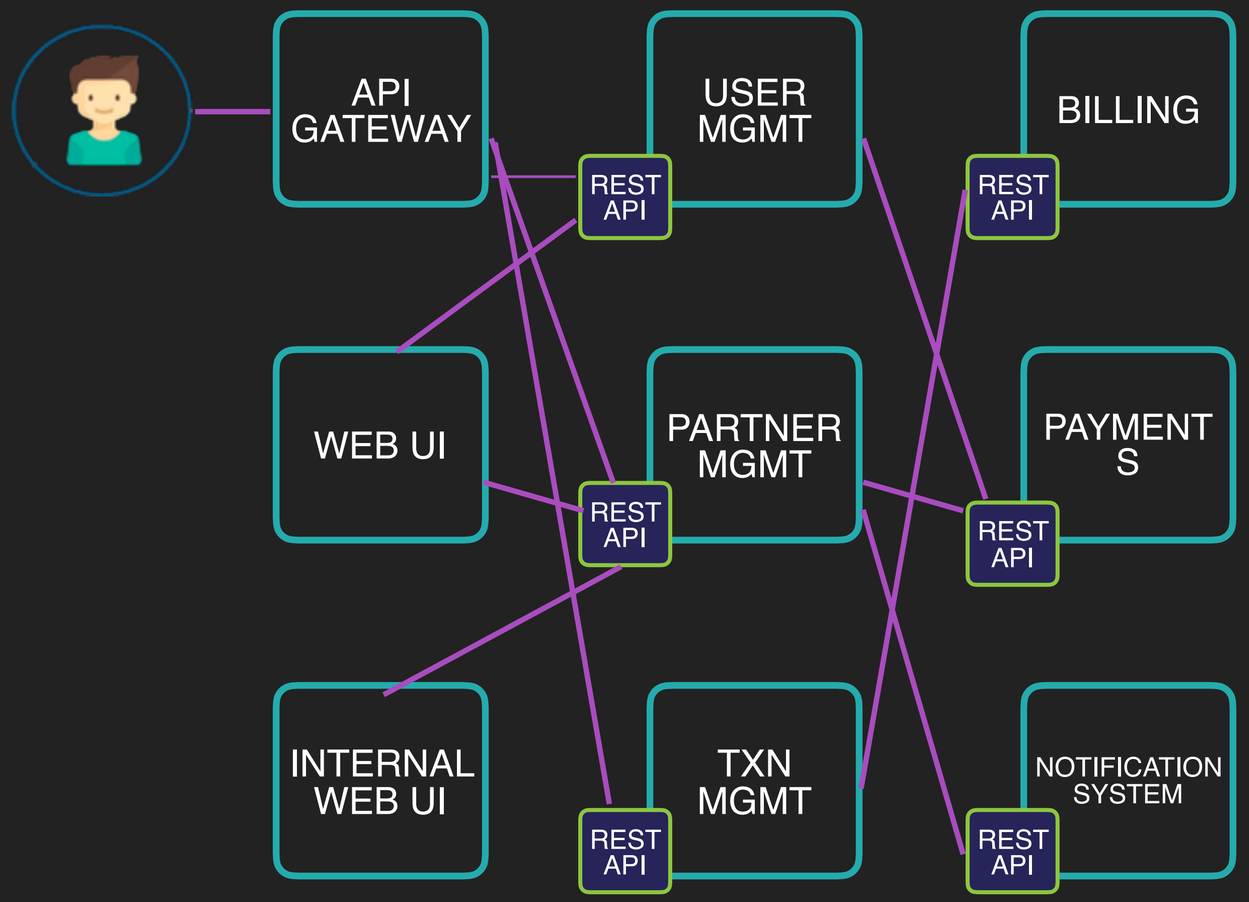
"Observation 1: Change is the most common trigger"



—Subbu Allamaraju, Expedia, Feb 2019
https://m.subbu.org/incidents-trends-from-the-trenches-e2f8497d52ed


APP

THEN


API GATEWAY
WEB UI
INTERNAL WEB UI
REST API
REST API
REST API
USER MGMT
PARTNER MGMT
TXN MGMT
REST API
REST API
REST API
BILLING
PAYMENTS
NOTIFICATION SYSTEM

NOW

DEV OPS

"Works on my machine"

"The only good diff is a red diff"

DEV OPS

THE FIRST WAVE: getting ops folks to code

THE SECOND WAVE: teaching devs to own code in production

observability

DEV OPS

it’s all about sharing

SOFTWARE OWNERSHIP

observability

a.k.a. understanding the behavior of a system based on knowledge of its external outputs.

a.k.a. "what is my software doing, and why is it behaving that way?"

monitoring

The system as black box magic. Thresholds, alerts, system signals like CPU and memory.

Checking and rechecking for known bad behaviors.

observability

The system as a living, adaptable thing. A culture of instrumentation and metadata rather than strictly-defined counters.

Being able to tease out previously-unknown bad behaviors and outliers.

DEV OPS

WRITE → TEST → COMMIT → RELEASE

💥 → DEBUG → FIX

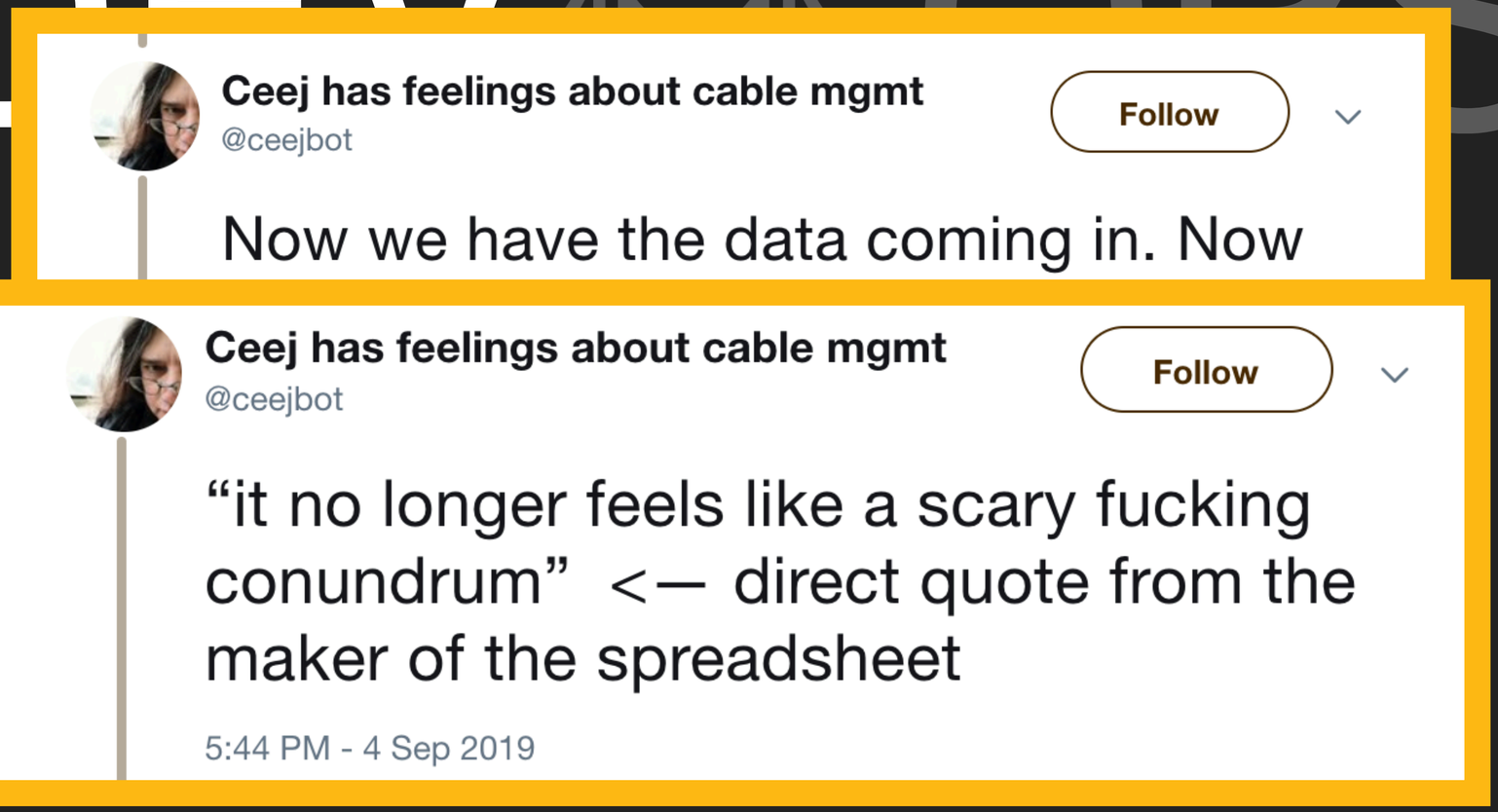
DEV OPS

WRITE → TEST → COMMIT → RELEASE → OBSERVE

TEST

OBSERVE

DEV ^ ^ OPS


Ceej has feelings about cable mgmt
@ceejbot
Follow
Now we have the data coming in. Now
Ceej has feelings about cable mgmt
@ceejbot
Follow
“it no longer feels like a scary fucking conundrum” <— direct quote from the maker of the spreadsheet
5:44 PM - 4 Sep 2019

… why devs, again?

The Software DEV Process

- Design documents
- Architecture review
- Test-driven development
- Integration tests
- Code review
- Continuous integration
- Continuous deployment
- 🎉🥂🍾🎊
- Observe our code in production

TEST

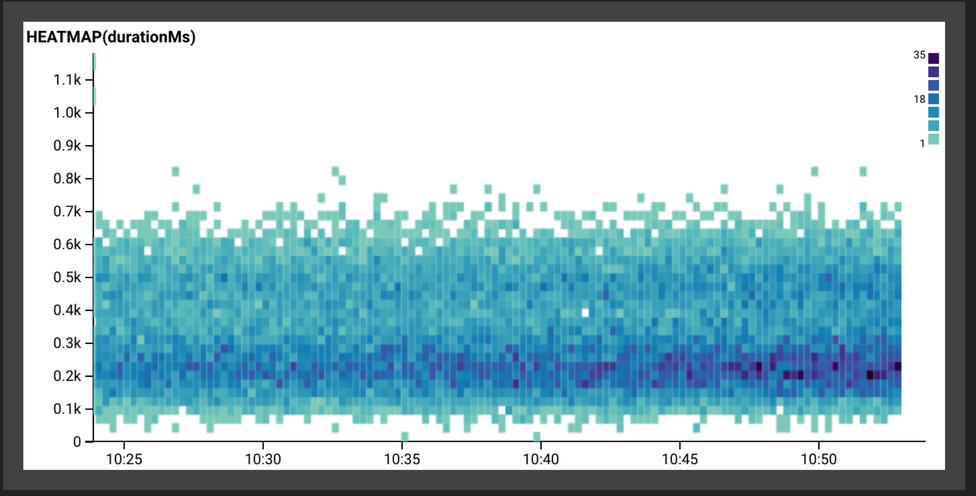
--- FAIL: TestUnitTest (0.00s)

talk_test.go:10: —

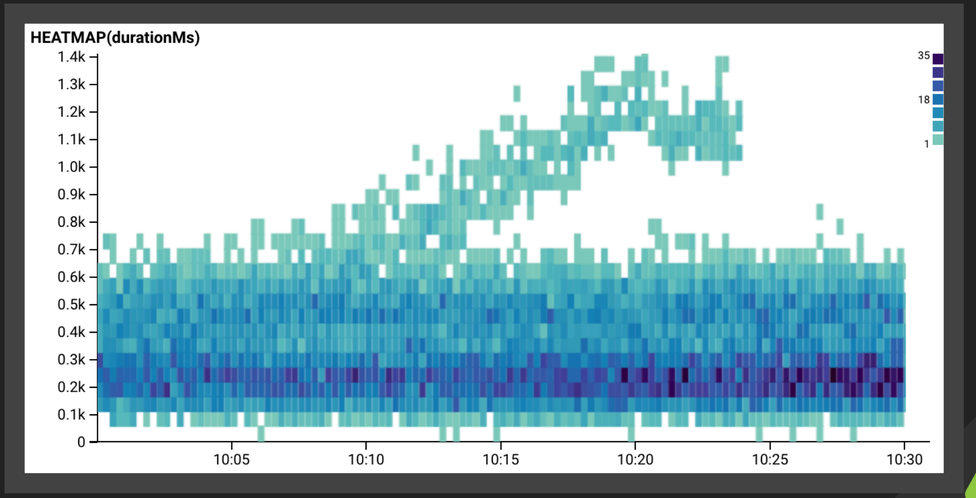
expected: 4 (type int)

actual: 5 (type int)

EXPECTED



ACTUAL



DEV 💥 OPS

"Works on my machine"

"The only good diff is a red diff"

DEV PROD

still
observability

prod, part of the dev process?

The Software DEV Process

- Design documents
- Architecture review
- Test-driven development
- Integration tests
- Code review
- Continuous integration
- Continuous deployment
- 🎉🥂🍾🎊
- (Wait for exception tracker to complain)

when deciding…

WHAT to build

HOW TO build it

WHETHER it works ("test in prod")

when deciding

WHAT

- Locally: log lines, printfs, debuggers attached to our IDEs
- What’s causing our code to deviate from expectations?
  - Stop "pulling straws"—quantify pain, and start prioritizing.

eaze

when deciding

HOW TO

- Know what "normal" really is
- Events (instrumentation) can be like DEBUG statements in prod
- What and how we build should be informed by reality

- Complex systems have an infinitely long list of black swan failure scenarios
- "Test in Production" to experiment and check hypotheses
- Feature flags + observability = 💛

when deciding

WHETHER

but this

is hard.

make prod feel more like dev

TOOLS SHOULD SPEAK MY LANGUAGE

- As a dev, traditional monitoring tools don't tie back to the concepts I deal with in my code

$YOUR_BIZ-relevant ID

AWS availability zone

time to render

API endpoint

CPU utilization

payload size

kafka partition

build ID

client OS

Cassandra hostname

TOOLS SHOULD SPEAK MY LANGUAGE

- As a dev, traditional monitoring tools don't tie back to the concepts I deal with in my code

AWS availability zone

us-east-1

eu-west-1

us-west-2

eu-central-1

us-west-1

customer ID

4e4e1207

BREAK DOWN
build_id

CALCULATE PER GROUP
COUNT
COUNT_DISTINCT(server_hostnam

FILTER
None; include all rows

ORDER
COUNT desc

LIMIT
10

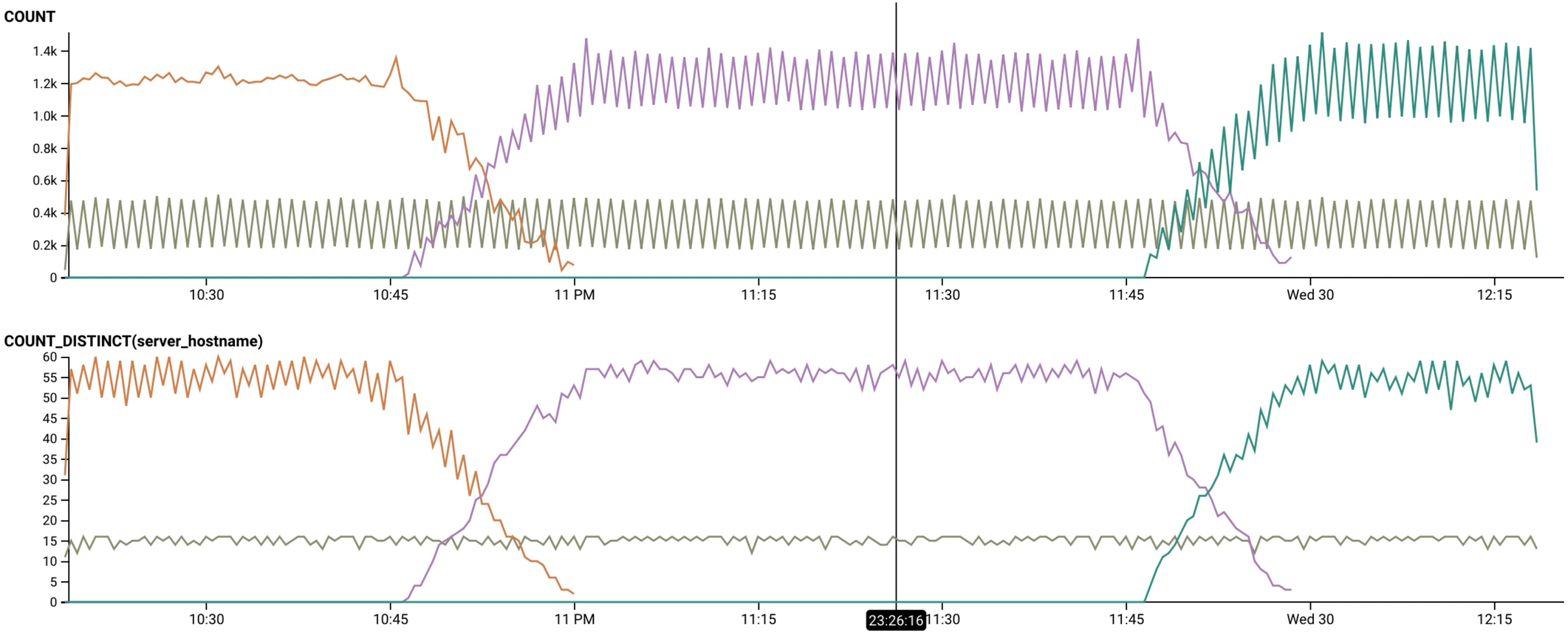
Run Query

Run 3 minutes ago

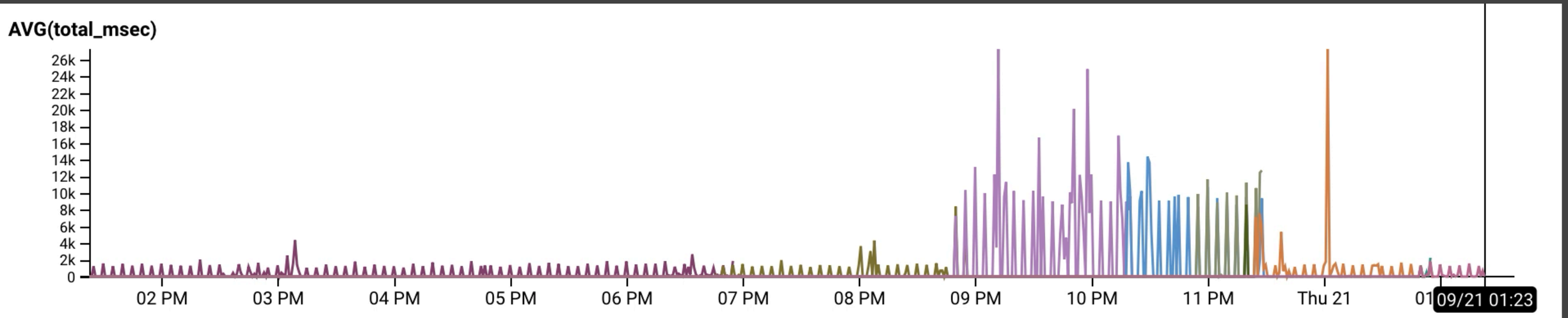
Jan 29 2019, 10:18 PM – Jan 30 2019, 12:18 AM

Results | BubbleUp New | Traces | Raw Data

Graph Settings



build_id	COUNT	COUNT_DISTINCT(server_hostname)
18017	144,388	60
18012	83,639	60



build_id	AVG(local_msec)	AVG(merge_msec)	AVG(total_msec)
4860	807.87638	22.21744	1,109.18379
4862	814.17609	18.75966	1,235.82825
4865	7,592.28437	16.59681	8,437.34694
4868	7,156.31961	12.81313	8,042.92072
4870	8,143.87158	11.72637	9,270.74609
4874	5,366.32319	6.33026	5,987.89468
4875	1,056.20032	20.95821	1,517.22261
4879	1,003.55562	32.23944	1,285.27912

TOOLS SHOULD SPEAK MY LANGUAGE

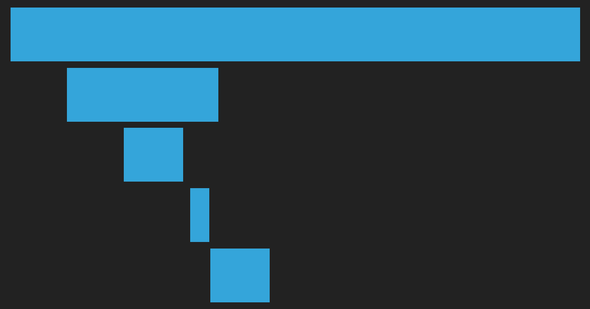
- As a dev, traditional monitoring tools don't tie back to the concepts I deal with in my code

AND LET ME ITERATE

SHARE PATTERNS WHERE POSSIBLE

- Tracing helps production feel even more familiar: can map a trace directly to my code structure

```
while len_p < nb_primes:
  for i in p[:len_p]:
    if n % i == 0:
      break
  else:
    p[len_p] = n
    len_p += 1
  n += 1
```



Untitled Query

Add to Board | Share | Last 2 hours

Datasets / retriever-traces

Add a description for this query

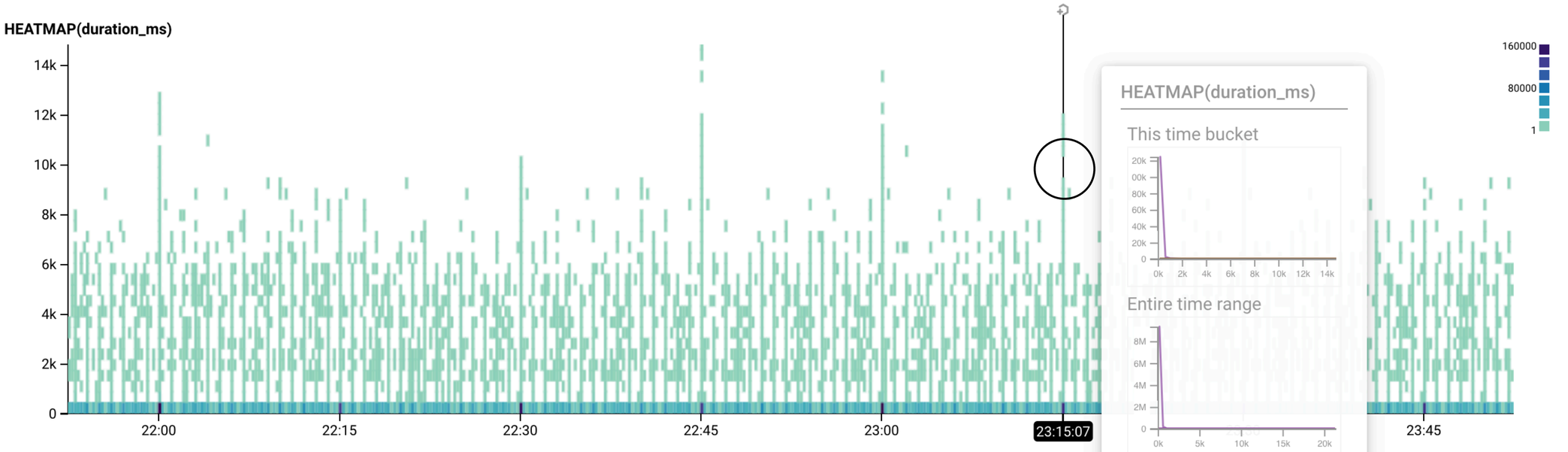
VISUALIZE	WHERE	GROUP BY	ORDER BY	LIMIT
HEATMAP(duration_ms)	None; include all rows	app.s3_prefix	None	None

Run Query

Run a few seconds ago

Sep 24 2019, 9:52 PM – Sep 24 2019, 11:52 PM

Results | BubbleUp | Traces | Raw Data | Graph Settings



app.s3_prefix	HEATMAP(duratic
d9e4-20151/70ef-17/a684-54	
d9e4-20151/cfcd-0/33ce-1213	

← Trace a50900bd-ba64-41f9-bb25-bd7783ac04cc at 2019-05-24 12:07:23

Search spans

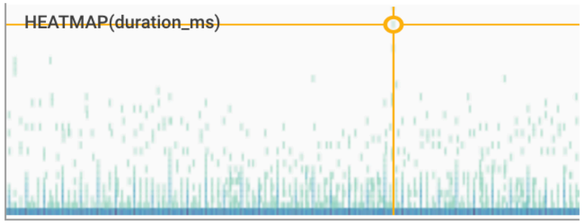
Fields

name	service_name	0s 10s 20s 30s 40s 50s 55.16
1 /api.RetrieverService/Fetch	poodle	55.148s
3 /api.RetrieverService/Fetch	retriever	55.147s
90 head.startQuery	retriever	0.3451ms
3 head.fetchMerged	retriever	53.230s
43 FetchPartialMerge	retriever	27.086s
1 FetchPartialMerge	retriever	29.264s
43 merging	retriever	28.889s
mergeBucket	retriever	1.311s
mergeBucket	retriever	401.0ms
mergeBucket	retriever	388.3ms
mergeBucket	retriever	398.7ms
mergeBucket	retriever	385.0ms
mergeBucket	retriever	962.2ms
mergeBucket	retriever	785.1ms
mergeBucket	retriever	809.1ms
mergeBucket	retriever	793.4ms
mergeBucket	retriever	828.4ms
mergeBucket	retriever	841.4ms

retriever >

FetchPartialMerge

Distribution of span duration



Not grouped

name = FetchPartialMerge
service_name = retriever

Fields

Filter fields and values in span

duration_ms
27086.312459

global.availability_zone
us-east-1d

global.build_id
110870

global.env
production

global.infra_type
aws_instance

global.instance_type

CHANGE CAN BE INCREMENTAL

2019-01-25T01:30:23.743Z Enqueued task

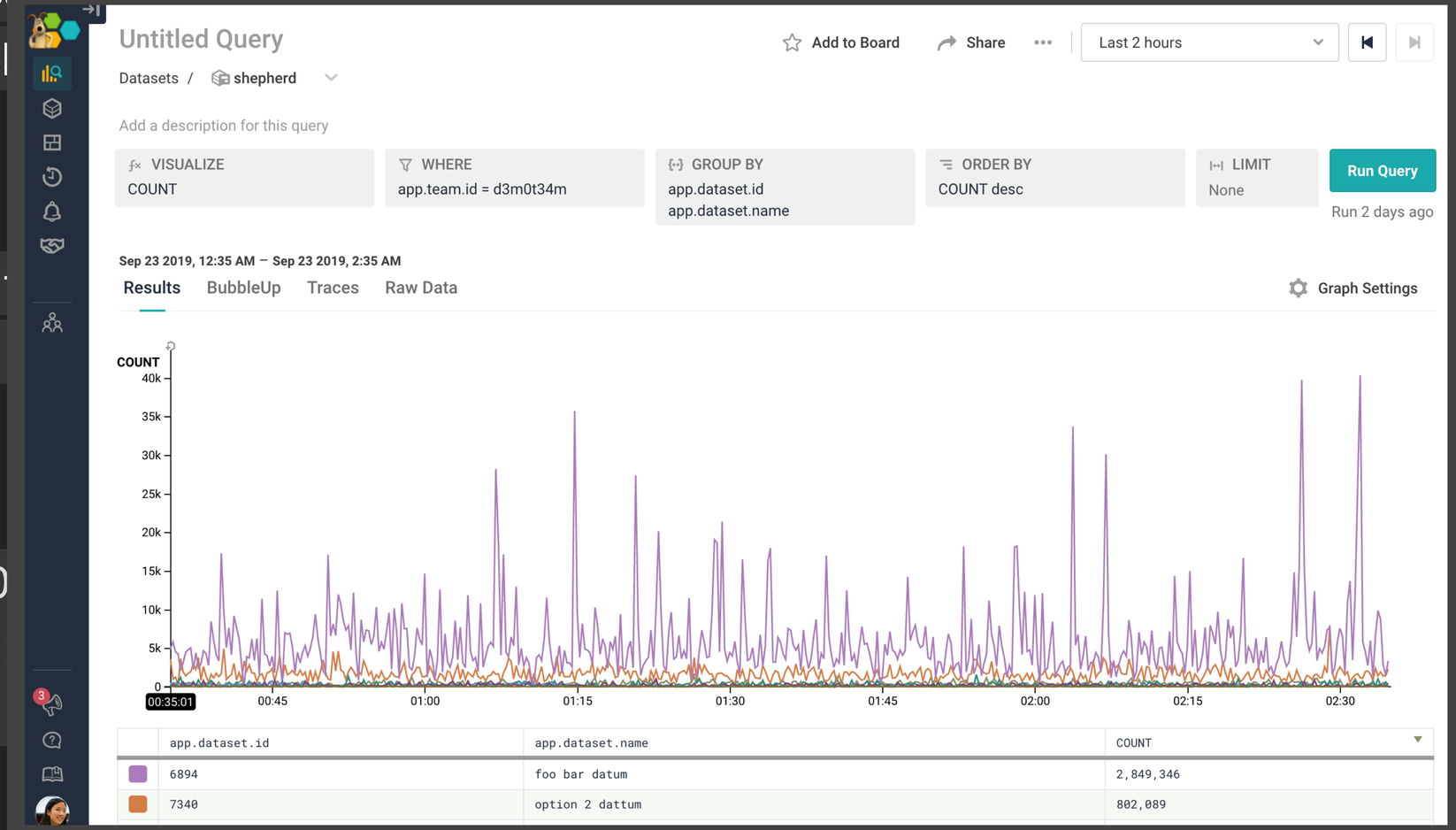
2019-01-25T01:30:24.120Z Task proce

2019-01-25T01:30:24.212Z Task compl

2019-01-25T01:30:23.743Z Enqueued

2019-01-25T01:30:29.953Z Task timed

Timestamp=2019-01-25T0
message=Task timed out
task_id=72



CHANGE CAN BE INCREMENTAL

```
2019-01-25T01:30:23.743Z Enqueued task task=72
2019-01-25T01:30:24.120Z Enqueued task task=74
2019-01-25T01:30:24.212Z Task processed, returning 42 entries task=74
2019-01-25T01:30:26.014Z Task complete (email sent to foobar@example.com) task=74
2019-01-25T01:30:26.214Z Enqueued task task=77
2019-01-25T01:30:24.120Z Task errored: unknown constant ::Fixnum task=77
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds task=72
2019-01-25T01:30:32.762Z Enqueued task task=78
2019-01-25T01:30:34.243Z Task processed, returning 0 entries task=78
2019-01-25T01:30:34.243Z Task complete, (email sent to bazqux@example.com) task=78
```

at the end of all of this…

DEV

OPS

DEV

OPS

DEV OPS

WRITE → TEST → COMMIT → RELEASE → OBSERVE

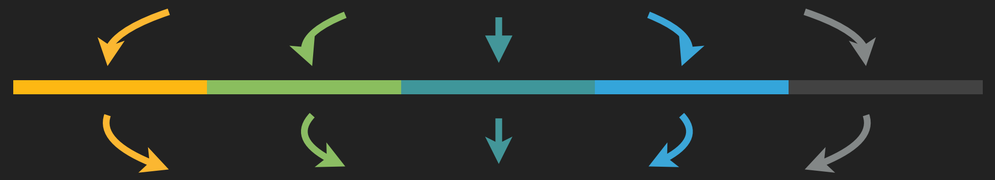
TEST

OBSERVE

OPS: share the great responsibility (and great power!)

DEVS: embrace observability, bring production closer to development.

ASK NEW QUESTIONS



SHIP BETTER SOFTWARE

@cyen
@honeycombio

CURIOUS? TRY play.honeycomb.io

We value your feedback!

Take the session survey through the PagerDuty Summit App!

Find Session > Session Feedback > Track Session Survey