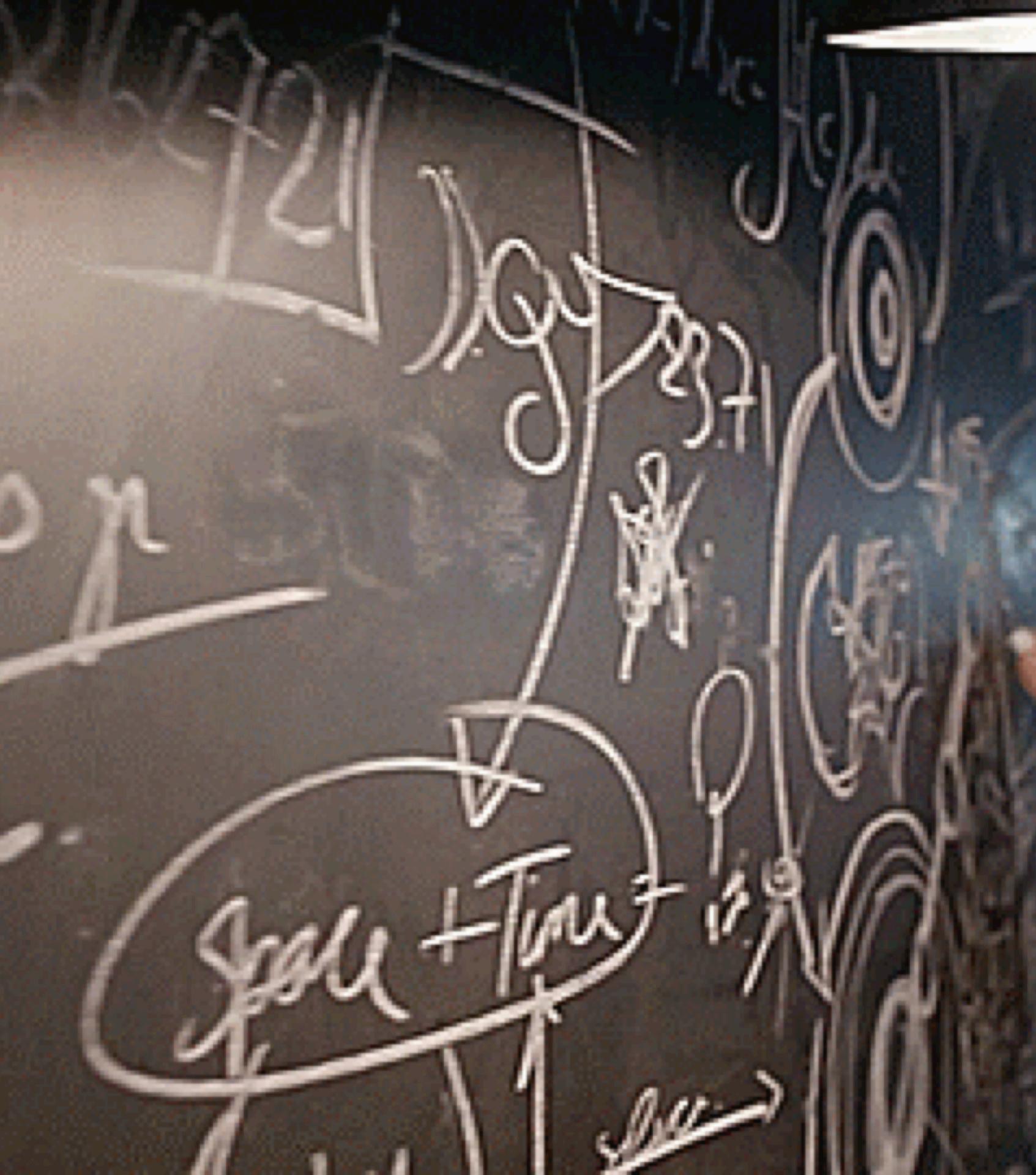


monitoring
on a budget

**a few animated gifs
with the Twelfth Doctor & 0 cats**



CJ Silverio
vp of engineering, **npm**
@ceejbot



let's talk npm
by the numbers

205 million packages Tuesday

10K requests/sec

npm is 25 people
4 of us run the registry

when the company was formed
5 people total

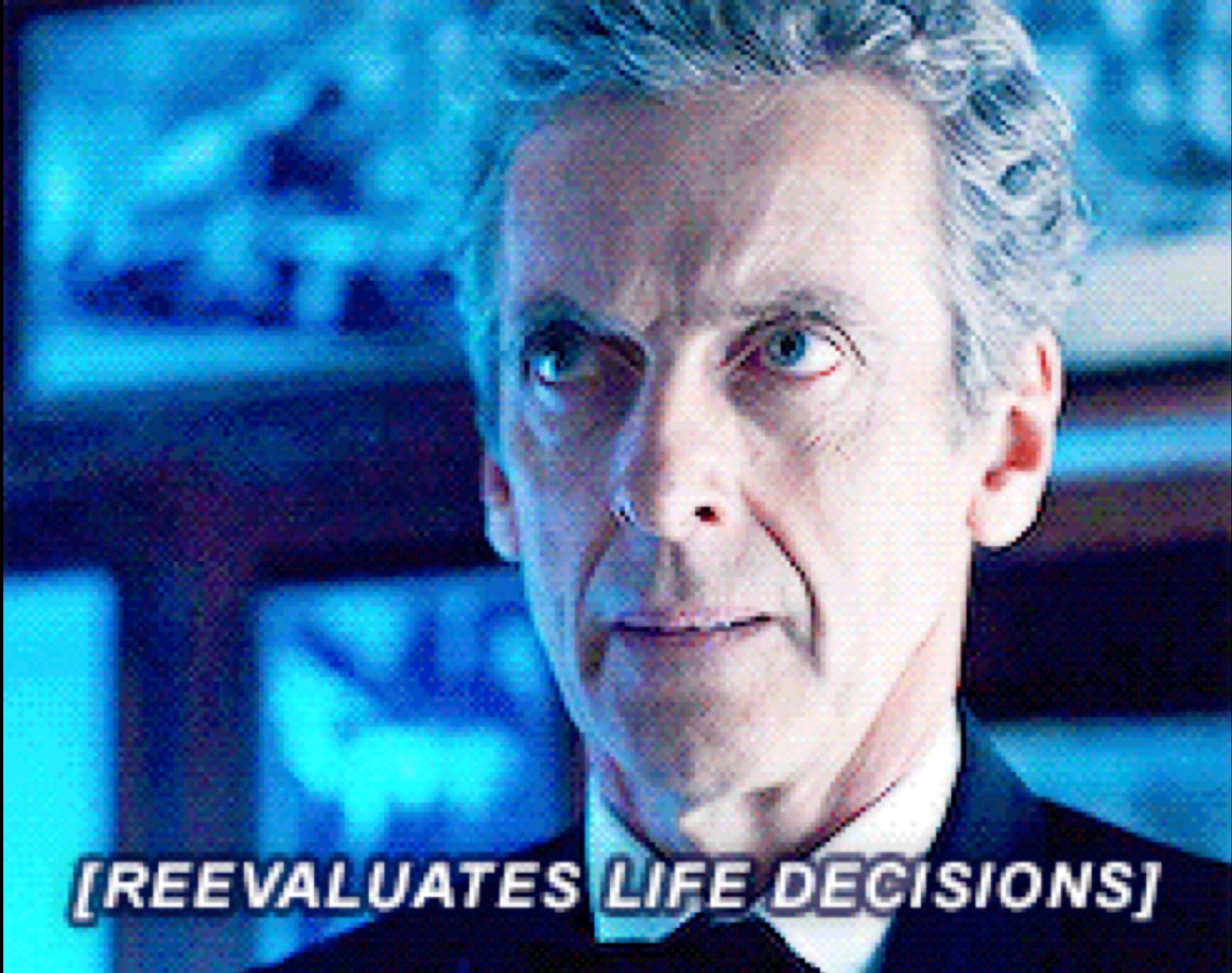
you outsource many services
when you're tiny

**you pull them back in-house
when you succeed**

**success is sometimes
a catastrophe**

npm's scale: runaway success
npm's staff: wouldn't this be neat

**mission: fix it
on a budget**



[REEVALUATES LIFE DECISIONS]

2 questions:
is the registry up?
how well is it performing?

is the registry up?
monitoring

how well is it performing?

metrics

monitoring

monitoring == pull
ask questions that you
know the right answers for

Is this host up?

Is this cert about to expire?

Is the DB replication keeping up?

**if you get the wrong answer
somebody gets paged**

nagios

state of the art in free

Current Network Status

Last Updated: Thu Feb 11 23:53:07 UTC 2016
Updated every 90 seconds
Nagios® Core™ 4.0.6 - www.nagios.org
Logged in as nagiosadmin

[View Service Status Detail For All Host Groups](#)
[View Status Overview For All Host Groups](#)
[View Status Summary For All Host Groups](#)
[View Status Grid For All Host Groups](#)

General

[Home](#)
[Documentation](#)

Current Status

[Tactical Overview](#)

[Map](#)

[Hosts](#)

[Services](#)

[Host Groups](#)

[Summary](#)

[Grid](#)

[Service Groups](#)

[Summary](#)

[Grid](#)

[Problems](#)

[Services \(Unhandled\)](#)

[Hosts \(Unhandled\)](#)

[Network Outages](#)

Quick Search:

Reports

[Availability](#)

[Trends](#)

[Alerts](#)

[History](#)

[Summary](#)

[Histogram](#)

[Notifications](#)

[Event Log](#)

System

[Comments](#)

[Downtime](#)

[Process Info](#)

[Performance Info](#)

[Scheduling Queue](#)

[Configuration](#)

Host Status Totals

Up	Down	Unreachable	Pending
78	0	0	0
All Problems	All Types		
0	78		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
598	0	0	0	0
All Problems	All Types			
0	598			

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
access-cache-1-west	UP	02-11-2016 23:52:42	13d 6h 5m 25s	PING OK - Packet loss = 0%, RTA = 1.20 ms
apex-redirector	UP	02-11-2016 23:52:34	13d 6h 4m 47s	PING OK - Packet loss = 0%, RTA = 0.87 ms
billing-api-1-west	UP	02-11-2016 23:52:12	13d 6h 5m 24s	PING OK - Packet loss = 0%, RTA = 0.40 ms
db-loadbalancer-1-east	UP	02-11-2016 23:52:45	13d 6h 5m 44s	PING OK - Packet loss = 0%, RTA = 85.28 ms
db-loadbalancer-2-west	UP	02-11-2016 23:52:59	13d 6h 5m 23s	PING OK - Packet loss = 0%, RTA = 0.84 ms
docs-2-west	UP	02-11-2016 23:52:15	13d 6h 5m 22s	PING OK - Packet loss = 0%, RTA = 0.33 ms
downloads-3-west	UP	02-11-2016 23:52:52	13d 6h 5m 22s	PING OK - Packet loss = 0%, RTA = 0.26 ms
enterprise-4-west	UP	02-11-2016 23:52:26	13d 6h 5m 21s	PING OK - Packet loss = 0%, RTA = 0.41 ms
etcd-1-west	UP	02-11-2016 23:52:24	13d 6h 5m 21s	PING OK - Packet loss = 0%, RTA = 0.38 ms
fastly	UP	02-11-2016 23:52:04	13d 6h 2m 50s	PING OK - Packet loss = 0%, RTA = 0.04 ms
fastly-purge-2	UP	02-11-2016 23:51:56	13d 6h 5m 20s	PING OK - Packet loss = 0%, RTA = 1.22 ms
fastlylogs-4-east	UP	02-11-2016 23:52:55	13d 6h 5m 20s	PING OK - Packet loss = 0%, RTA = 72.60 ms
frontdoor-1-eu	UP	02-11-2016 23:52:41	13d 6h 5m 19s	PING OK - Packet loss = 0%, RTA = 156.68 ms
frontdoor-10-east	UP	02-11-2016 23:52:21	8d 4h 13m 43s	PING OK - Packet loss = 0%, RTA = 83.70 ms
frontdoor-11-east	UP	02-11-2016 23:52:09	2d 20h 7m 29s	PING OK - Packet loss = 0%, RTA = 78.93 ms
frontdoor-12-west	UP	02-11-2016 23:52:42	7d 4h 15m 2s	PING OK - Packet loss = 0%, RTA = 0.30 ms
frontdoor-13-west	UP	02-11-2016 23:52:56	7d 4h 15m 1s	PING OK - Packet loss = 0%, RTA = 1.15 ms
frontdoor-6-west	UP	02-11-2016 23:51:58	13d 6h 5m 45s	PING OK - Packet loss = 0%, RTA = 1.27 ms
frontdoor-7-east	UP	02-11-2016 23:52:06	0d 3h 25m 47s	PING OK - Packet loss = 0%, RTA = 69.81 ms
frontdoor-8-west	UP	02-11-2016 23:51:56	9d 1h 25m 23s	PING OK - Packet loss = 0%, RTA = 0.42 ms
frontdoor-9-east	UP	02-11-2016 23:52:51	9d 0h 22m 29s	PING OK - Packet loss = 0%, RTA = 74.75 ms
frontdoor-canary-2	UP	02-11-2016 23:52:29	13d 6h 6m 1s	PING OK - Packet loss = 0%, RTA = 0.82 ms



One

**It's okay. We never look at it.
It just triggers Pager Duty.**

**nagios's virtues:
reliability & custom checks**

goal: never page anybody

self-healing checks
automate the fix if you can!

monitoring == unit tests
a ratchet for continuous improvement

external monitoring
ping services

**you must monitor
but that's just the start**

**monitoring tells you what
it doesn't tell you why**

metrics

Q: What's a metric?

A: A name + a value + a time.

counter: it happened N times

gauge: it's Y-sized right now

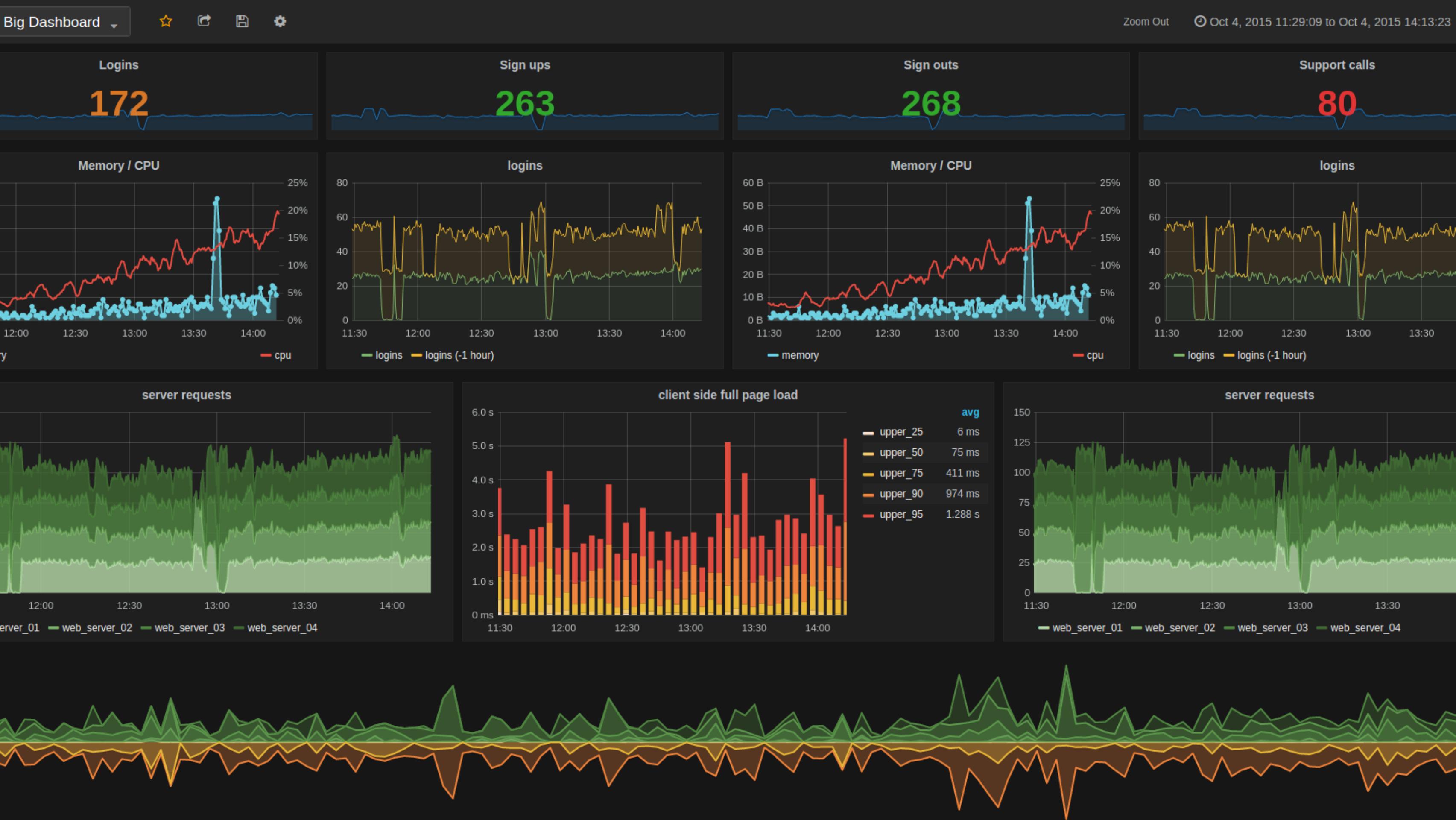
rate: it's happening N times/second

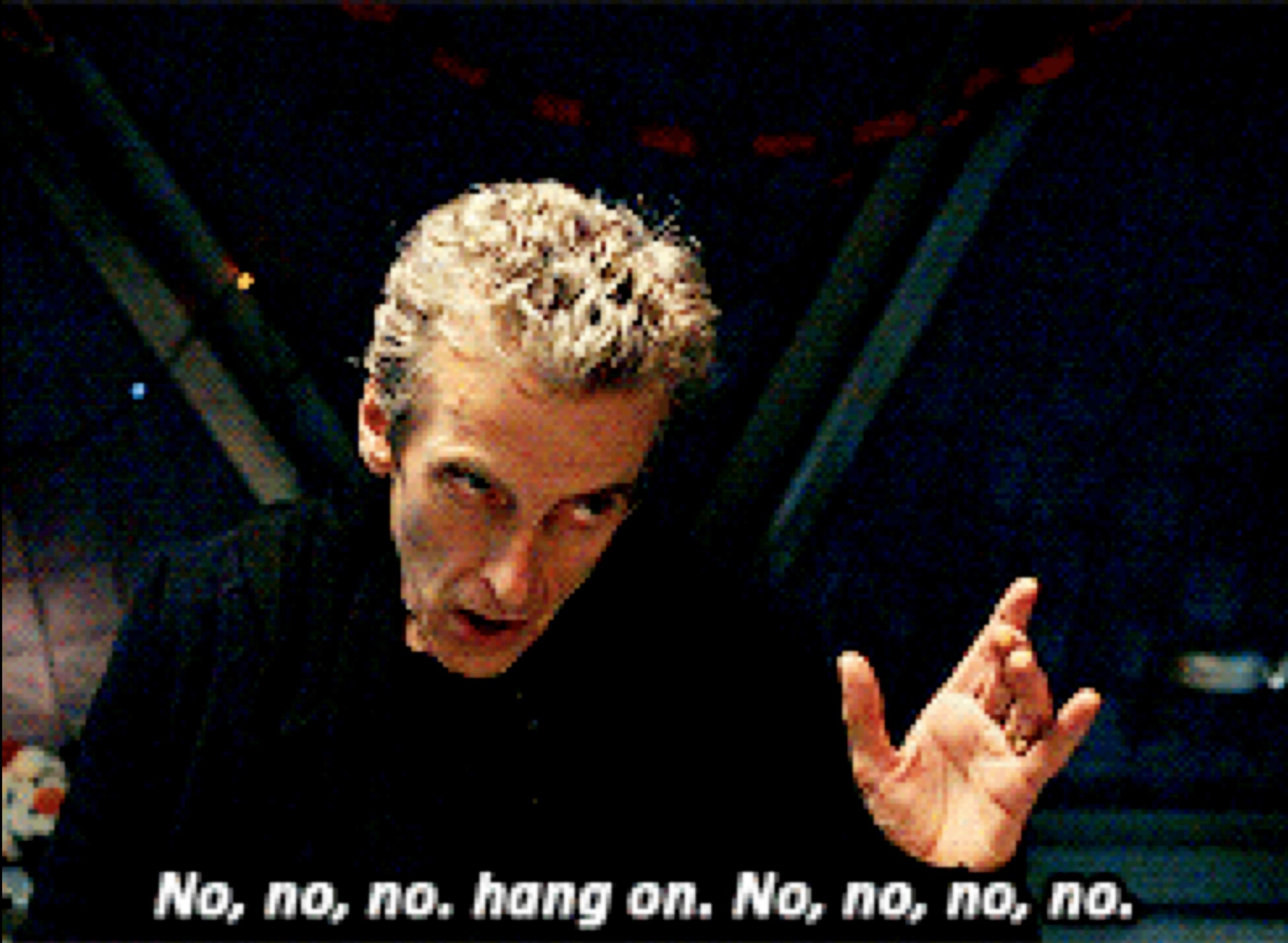
timing: it took X milliseconds

metrics ≡ **push**
the app gives you numbers

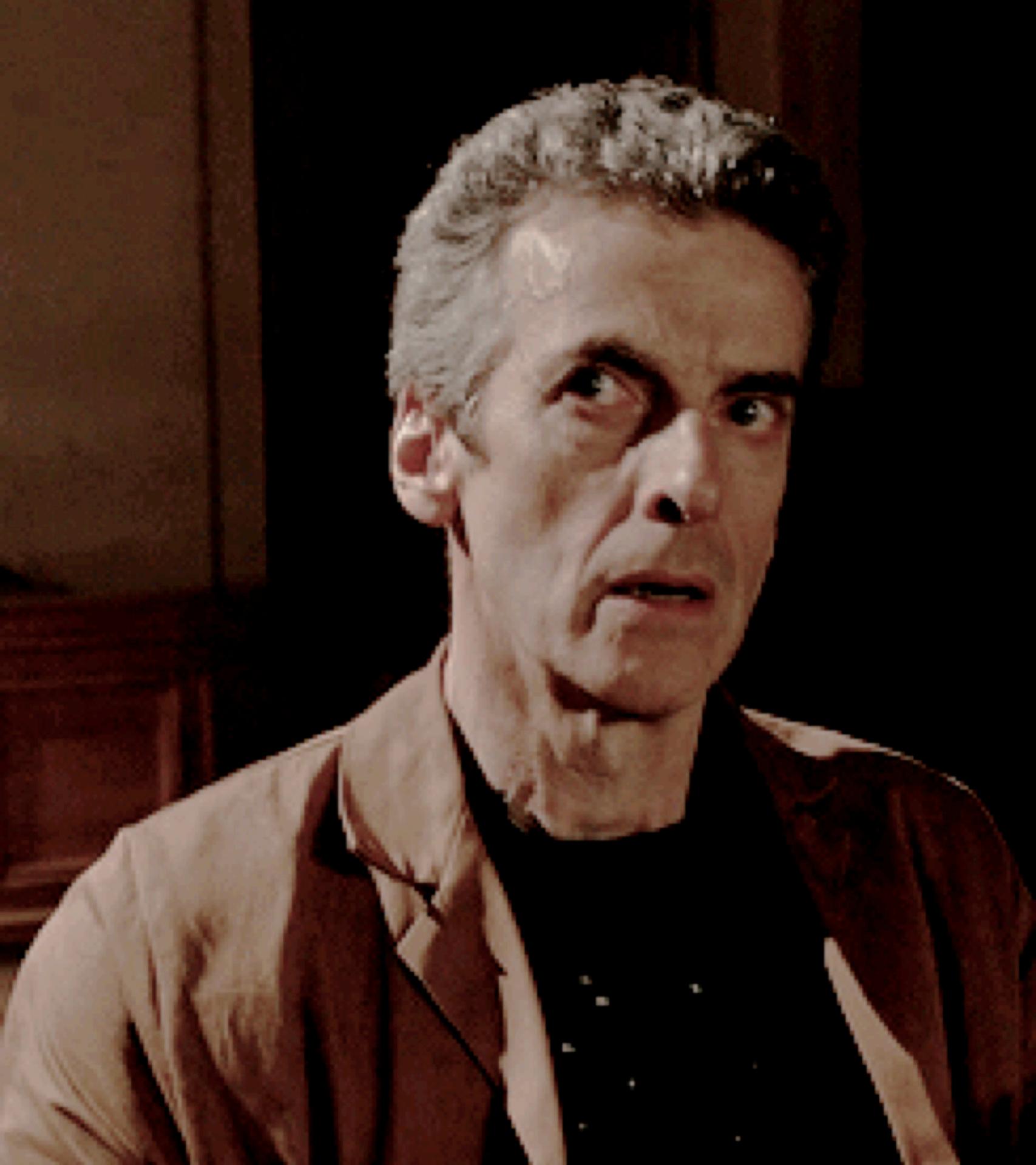
**emit from a service
store in timeseries db
query & graph**

the usual stack
statsd → graphite → grafana





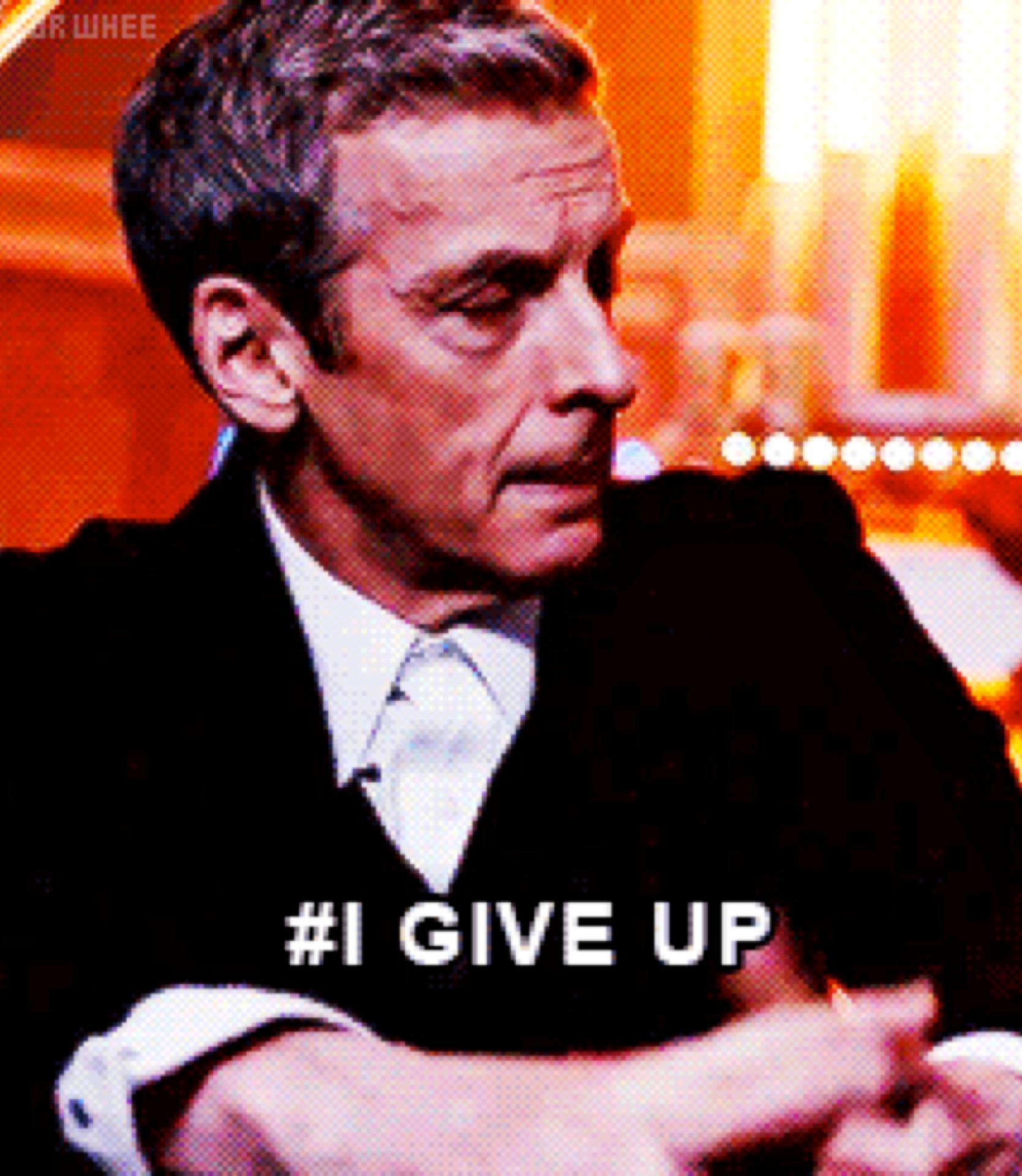
No, no, no. hang on. No, no, no, no.

A portrait of a middle-aged man with white hair, wearing a light-colored button-down shirt. He has a serious, slightly weary expression and is looking off to the right of the frame. The background is dark and indistinct.

**statsd uses
UDP**

Q: Why not send metrics over UDP?

A: You care about receiving them.



#I GIVE UP

just try to
install
graphite

**for-pay/SaaS services exist
but I can't afford them**

monitoring 400 processes right now
12+ GB of log data a day

interlude:
when should you pay?

convert the £\$€ cost
into engineer hours/month

**pay when it's cheaper than
investing an engineer
(be honest about the cost)**

**numbat was born
“How hard can it be?” I said.**



#COME AT ME BRO!

[https://github.com/
numbat-metrics](https://github.com/numbat-metrics)



npm's stack

numbat → **influxdb** → grafana

```
var Emitter = require('numbat-emitter');

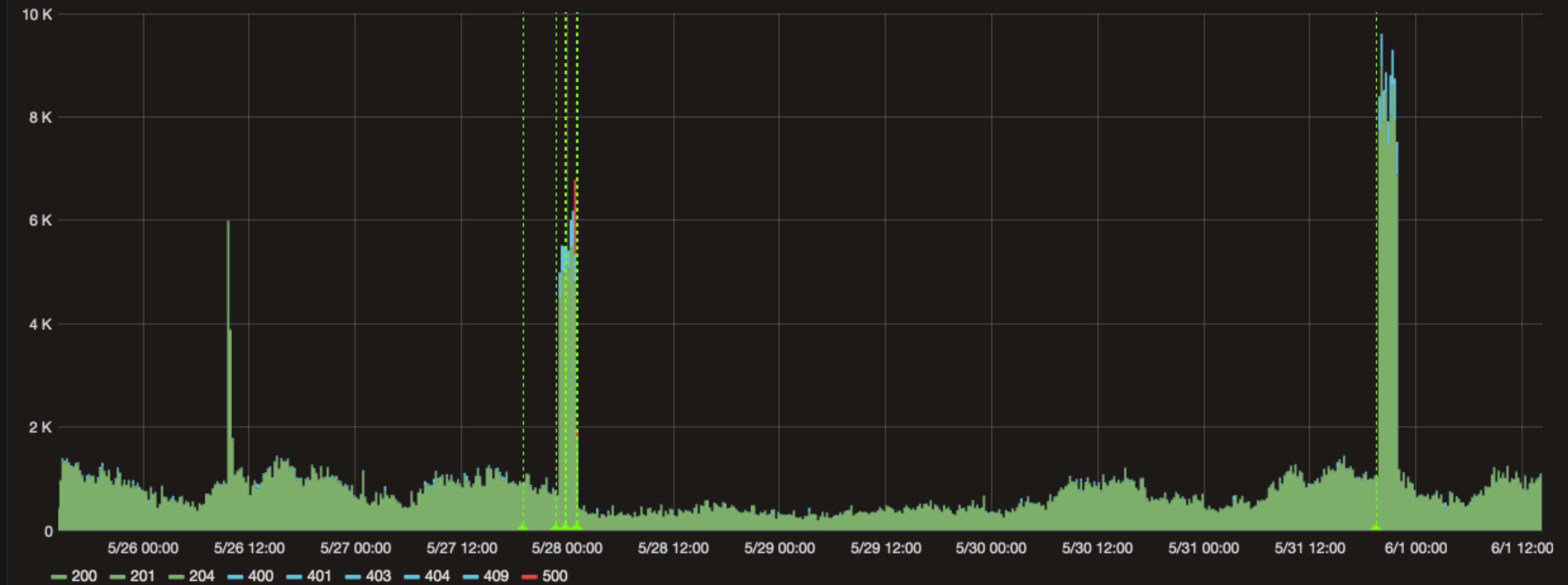
var emitter = new Emitter({
  uri: 'tcp://localhost:3333',
  app: 'www',
});

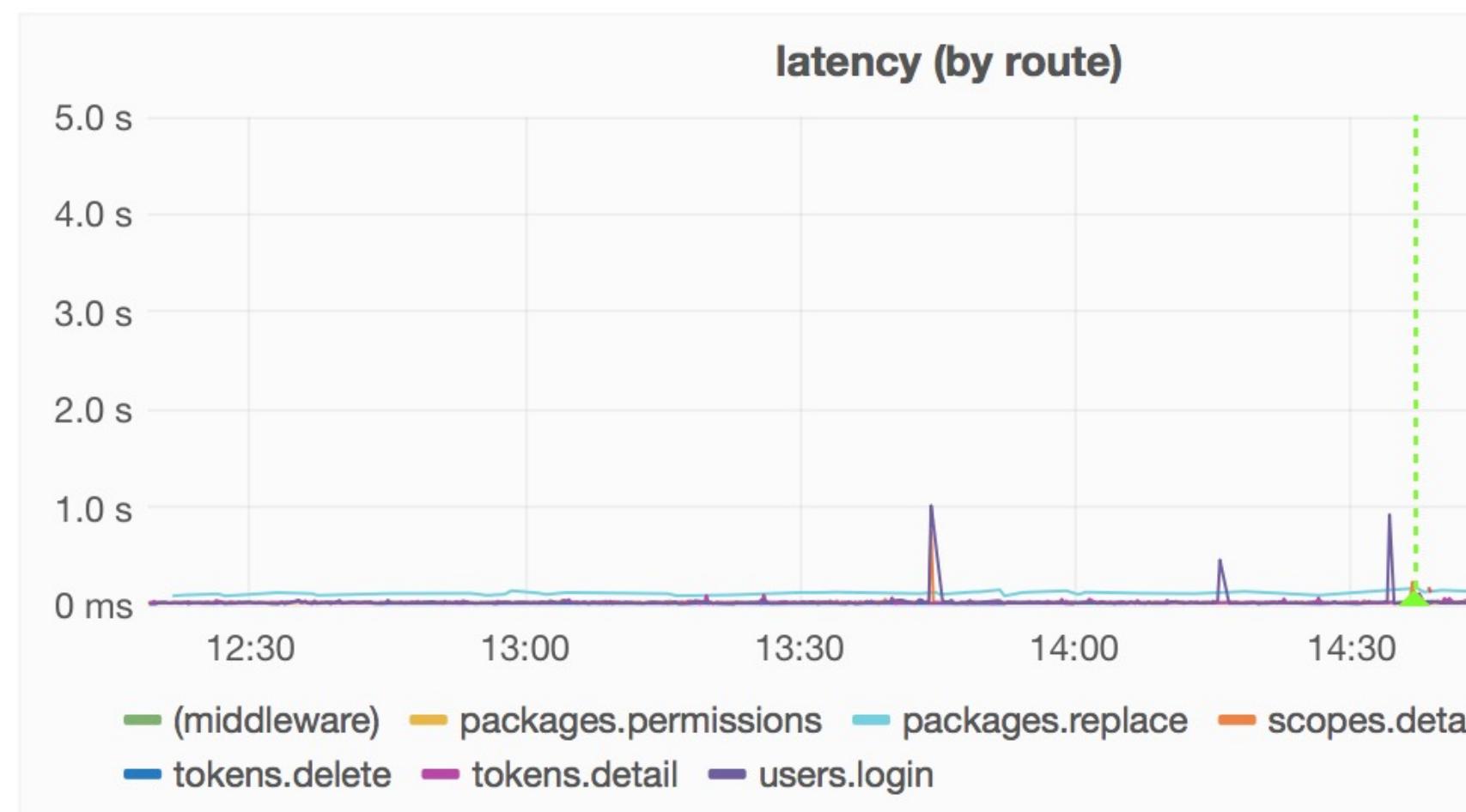
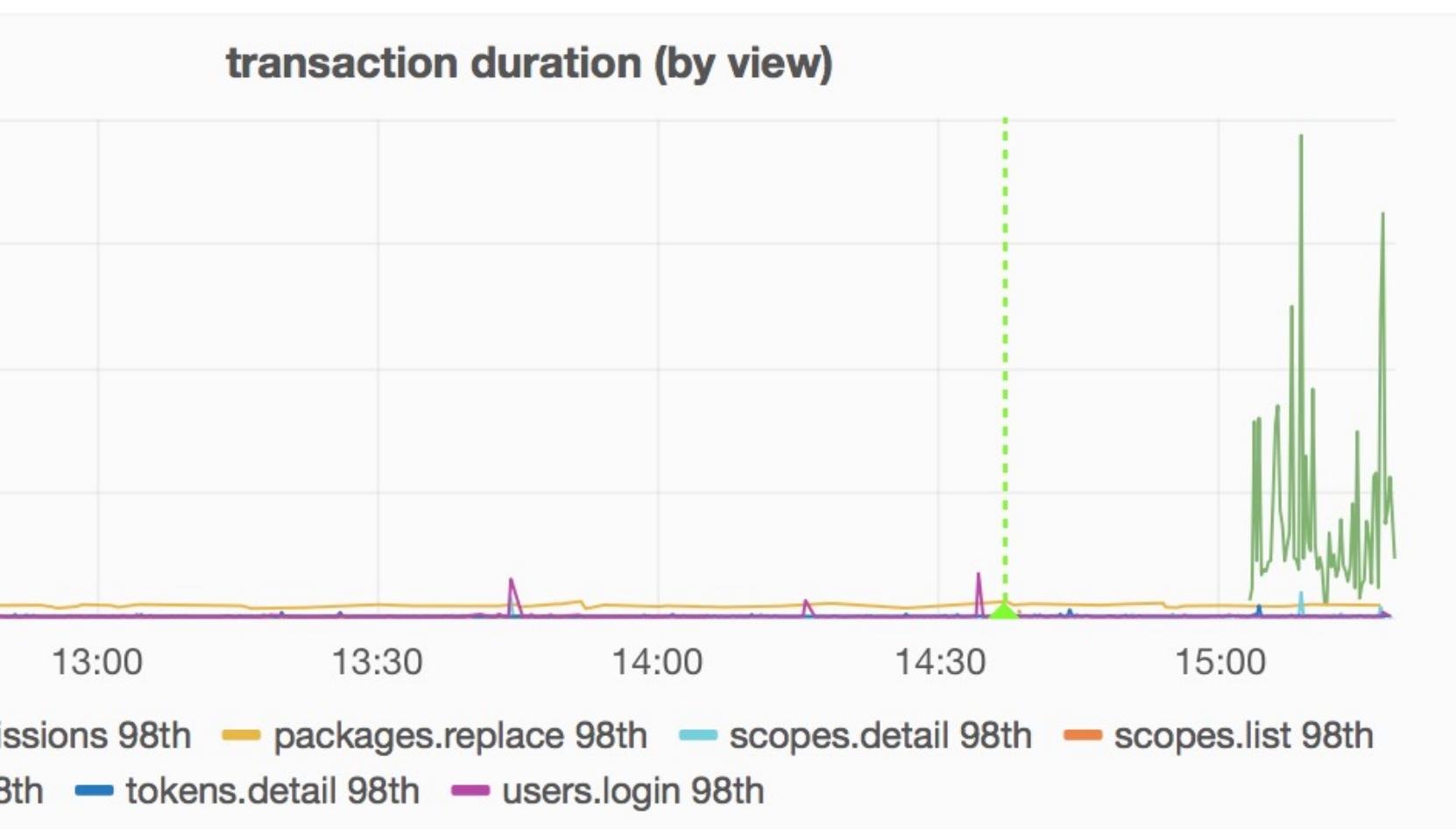
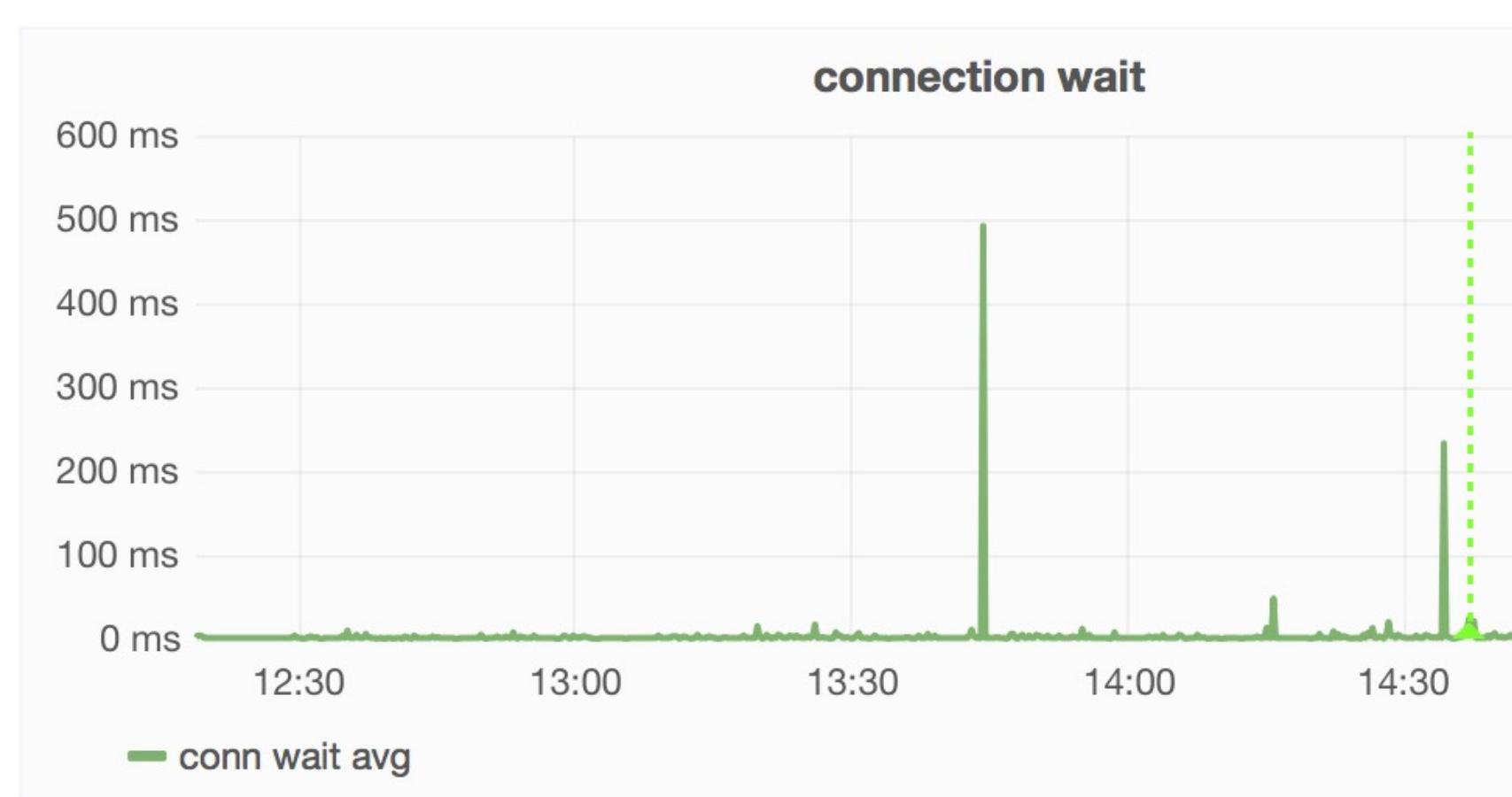
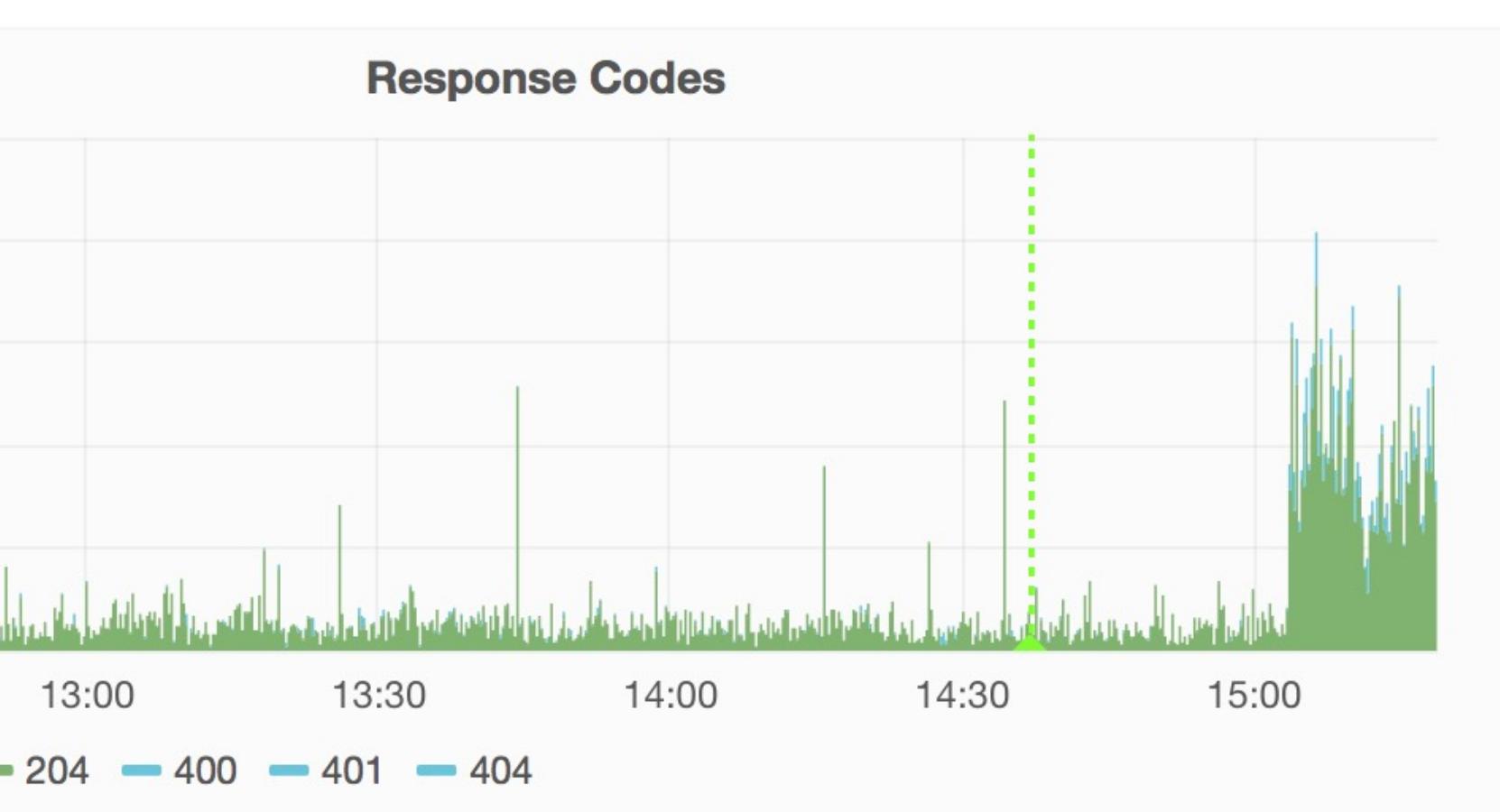
process.emit('metric', { name: 'request.latency', value: 30 });
process.emit('metric', { name: 'disk.used.percent', value: 36 });
process.emit('metric', { name: 'login' });
```

so easy to emit a metric
that we just do it any time
something interesting happens

**4000 metrics/sec
from the registry**

Response Codes

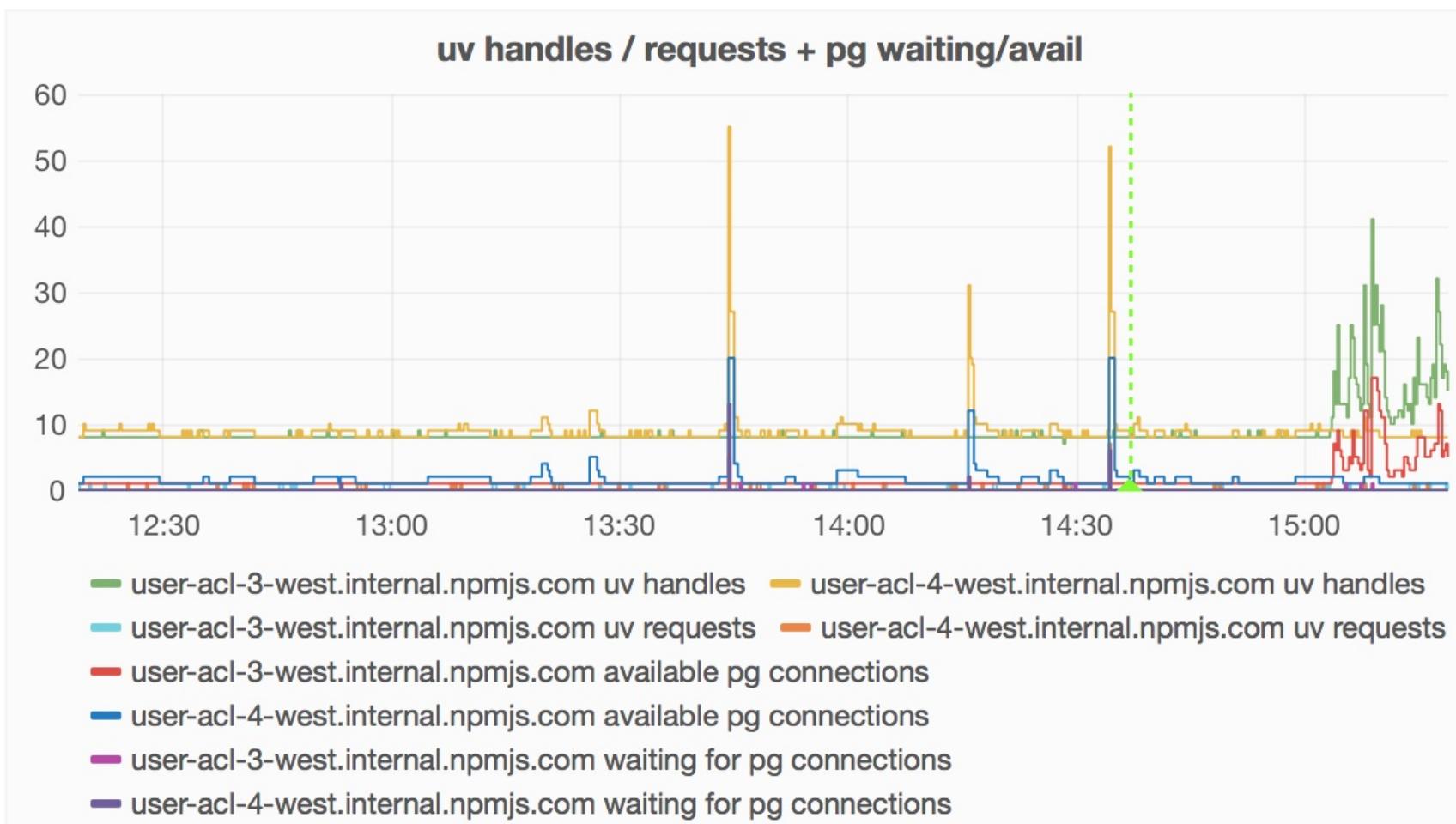
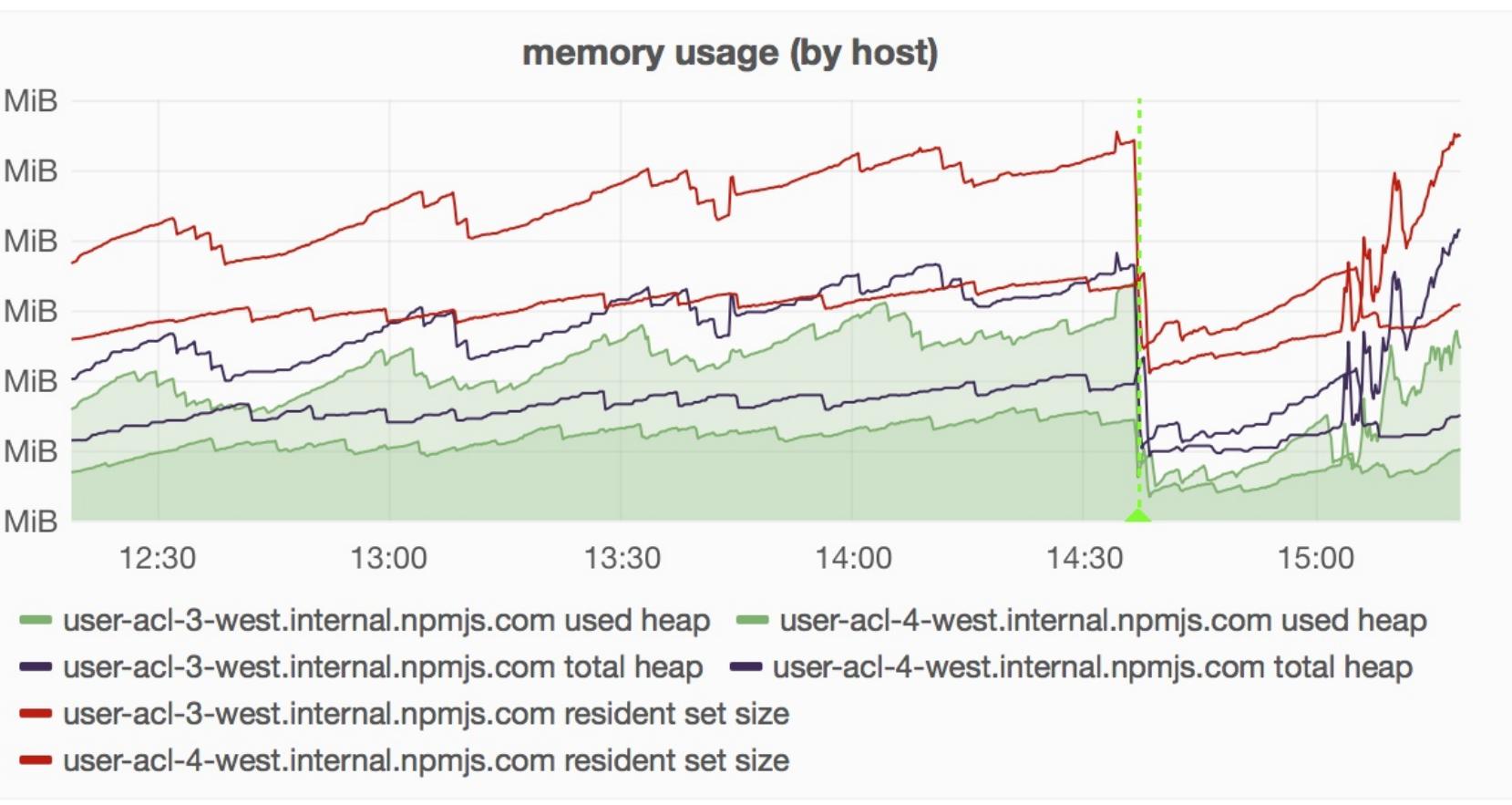
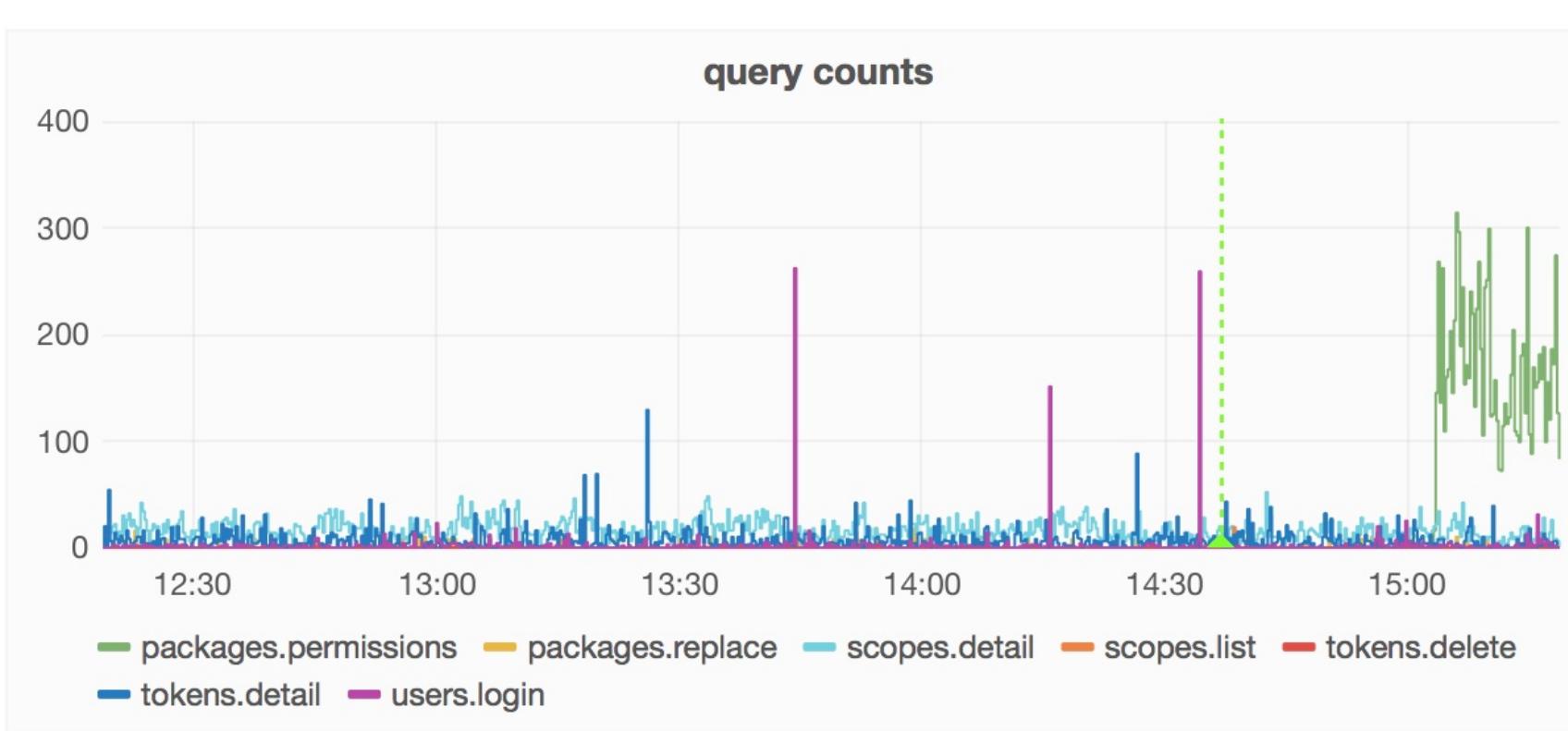
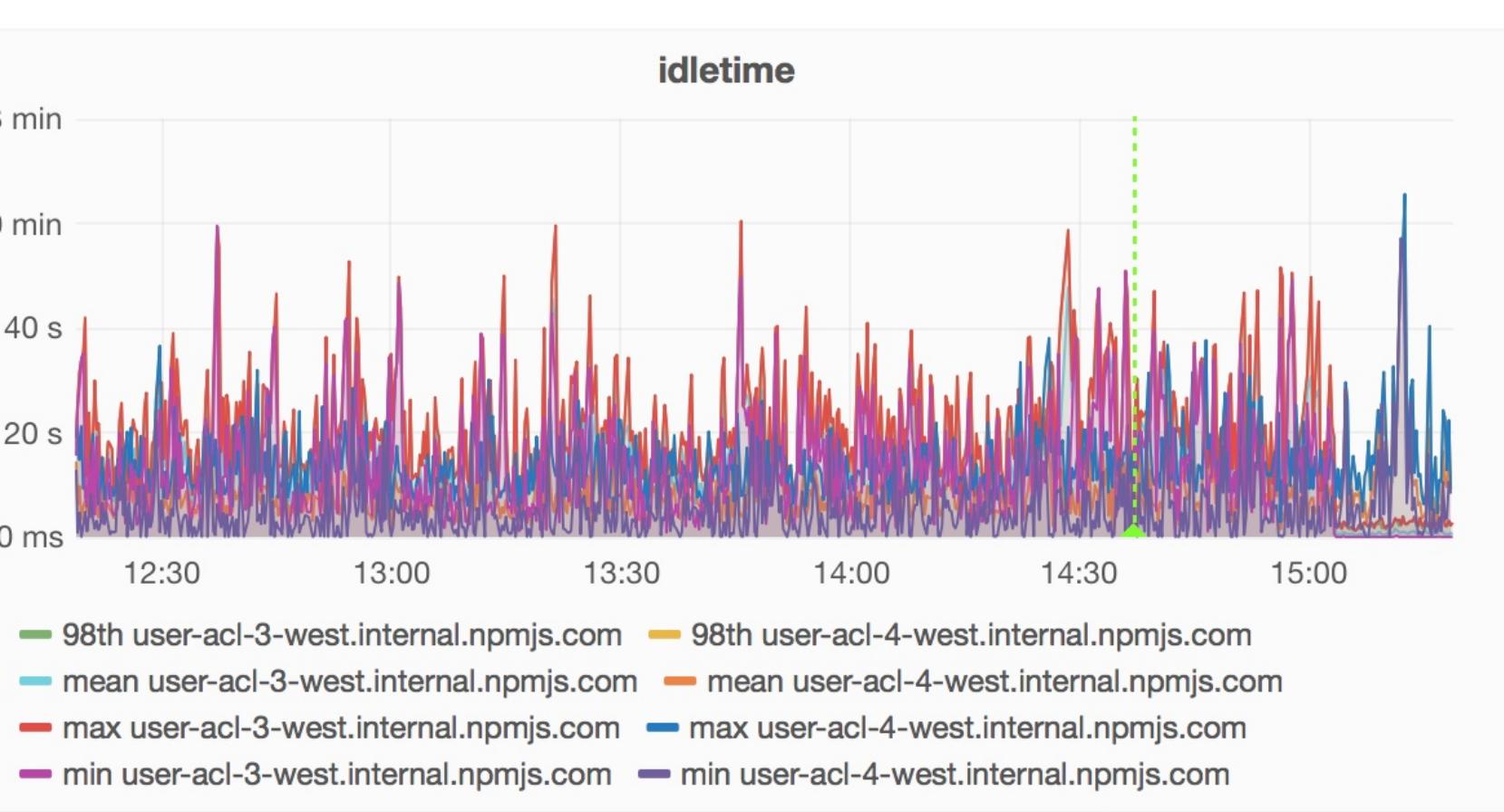






Oh, that can't
be good.

© 2002 THE CHILOIS GROUP



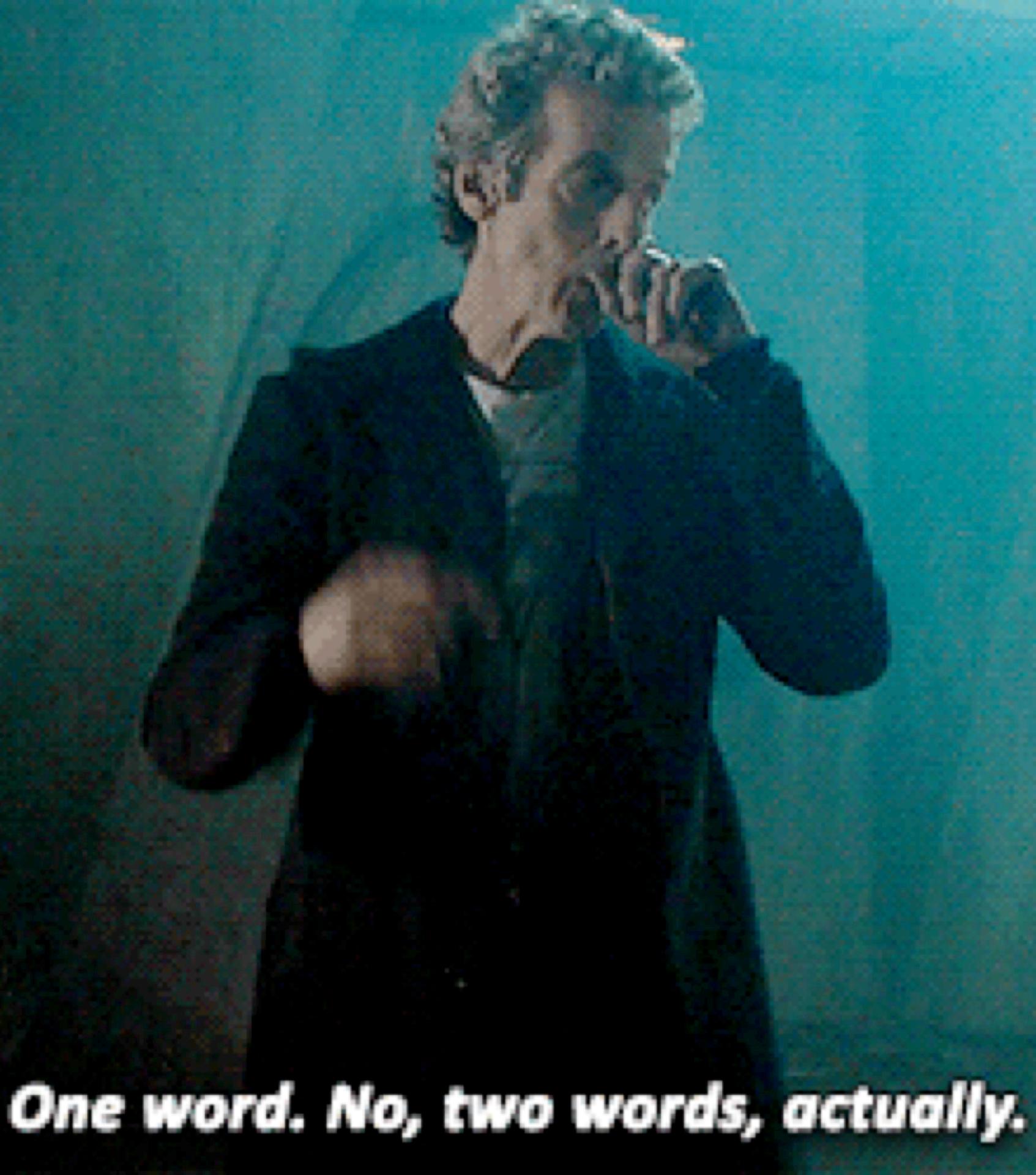
metrics → alerts

Server handling expected traffic?

Latency higher than normal?

Error rate higher than usual?

**metrics comprise a data stream
send the stream to more than one place!**



One word. No, two words, actually.

anomaly detection

**get data on what
your services are up to**

**what: monitoring
yes/no questions**

why: metrics
data changing over time

**next: anomaly detection
predictions & trends**

automate
don't require humans



npm install -g npm@latest
@ceejbot on all the things
npm loves you