

the accidental noder

**or, how I woke up & found myself running  
the npm registry**

**interspersed with diverse lessons  
learned through a series of epic mistakes**

**with the occasional juicy hint  
about upcoming npm projects**

the accidental noder  
CJ Silverio, @ceejbot

**what does a CTO do, anyway?**

npm's engineers and  
and the systems they build  
are under my care

**where they are now &  
where they're going next**

npm's **cli**: 3 people  
the tool you use to install

**npm's website: 4 people  
where you look up the packages**

**npm's registry: 5 people  
storing & serving 350K packages**

**record-scratch noise**

**Woo. What a job.**

**How many times have I done this before?**

**my background looks nothing like this**

how the **heck** did this happen?  
I didn't set out to do this.



let's go back in **time**

**I've been in the industry since 1989  
this means I'm old enough to be your mom**

**imagine me glaring next time somebody says  
your mom could do something**

I worked at a bunch of **forgotten** places  
**General Magic, WebTV, etc**



**Lesson: adapt or get selected out**

consolation prize: acquisition  
**funemployment!**

**modern web development**

**woo, ruby on rails + backbone!**

**ruby is too slow**

**node 0.4**  
**allegedly fast & v hipster**

**0.4 was barely usable, but  
async i/o was what I wanted**

**javascript was a surprise:  
fast w/ first-class functions**

**npm existed but was different**  
**you found modules by reading a wiki page**

- [sass.js](#) — Parses Sass templates and renders css.
- [scss.js](#) — Parses SCSS templates and renders css.
- [template.node.js](#) — A light, fast, cached template module for node.
- [tmpl-node](#) — a feature-rich template module for node.js
- [jsdom](#) — pure js implementation of the dom level 1 with some browser augmentation. Level 2 and 3 are being considered.
- [bind-js](#) — a simple templating engine for node.js that smiles back.
- [normal-template](#) — Normal templates are simple, yet powerful. They are safe, usable in non XML/HTML contexts and portable to any programming language.
- [nun](#) — Totally asynchronous non-blocking template engine for node.js
- [node.magic\\_dom](#) — A DSL for building HTML in node.js, similar to Python's Stan
- [stencil](#) — Renders asynchronous templates shared by server and browser
- [strobe-templates](#) — An asynchronous templating engine with syntax like Django's
- [nTPL](#) — Fast & Native extendable template system
- [CoffeeKup](#) — Markup as CoffeeScript. (sep 2010, node 0.2.0)

## Build and Deployment

- [nodejs-autorestart](#) — It is a way to watch all .js files if they have been changed and to restart nodejs. It allows easy development and stable production.
- [Jake](#) — JavaScript build tool similar to Make or Rake.

## Package Management Systems

- [kiwi](#) — Feature rich, fast, node.js package management system — server sponsored by SliceHost ([no longer in active development](#))
- [npm](#) — A node package manager that uses CommonJS-compatible package.json files, written in asynchronous javascript.
- [mode](#) — Distributed Node module repository. Uses a [github repository](#) to contain info about installable modules.
- [seed](#) — Universal package manager for CommonJS. Includes command line tools and async server

docs were terrible  
I had to learn by reading source

**node was the most fun  
I'd ever had writing code**

I found my **oldest** node project  
and tried to run it

# Woah. It runs.

```
info: ----- ljsnarf run started
info: Version: 0.1.3
info: Source account: ceej@www.livejournal.com
info: Generating session using challenge/response
debug: making flat API call with mode: sessiongenerate
debug: requesting fresh challenge
(node:87517) DeprecationWarning: util.puts is deprecated. Use console.log instead.
(node:87517) DeprecationWarning: util.error is deprecated. Use console.error instead.
info: Recording userpic keyword info for ceej
info: Read cached data from userpics.json.
debug: making flat API call with mode: login
```

**Voxer: a walkie-talkie app  
with a 100% node back end**

**0.6 was the first node  
that actually worked**

attempting to evangelize node  
among people writing Scala

at the end of 2013  
npm started falling over

**Lesson: in a viral success  
costs outscale goodwill**

@izs contacted me  
do devops for npm?

heck yeah



The Spice must flow.

we got it **stable** then we  
rewrote it in node & made it **scale**

**2014 was a good year for node**

node 0.10 was stable  
npm became stable

# April 29, 2014



**seldo** 20:23

New daily downloads record! 10,204,076

# October 25, 2016



**seldo** 20:08

Record downloads day: 286,777,458

**but wait! that was a month ago!**



**ceej** 10:33

hermione: npm downloads

---



**hermione** BOT 10:33

npm downloads by day:

2016-11-13: 84,436,813

2016-11-14: 265,069,276

2016-11-15: 296,334,144

2016-11-16: 293,541,433

2016-11-17: 292,932,490

2016-11-18: 258,965,189

2016-11-19: 90,858,568

**Total:** 1,582,137,913

**296,334,144**

**packages installed last Tuesday**

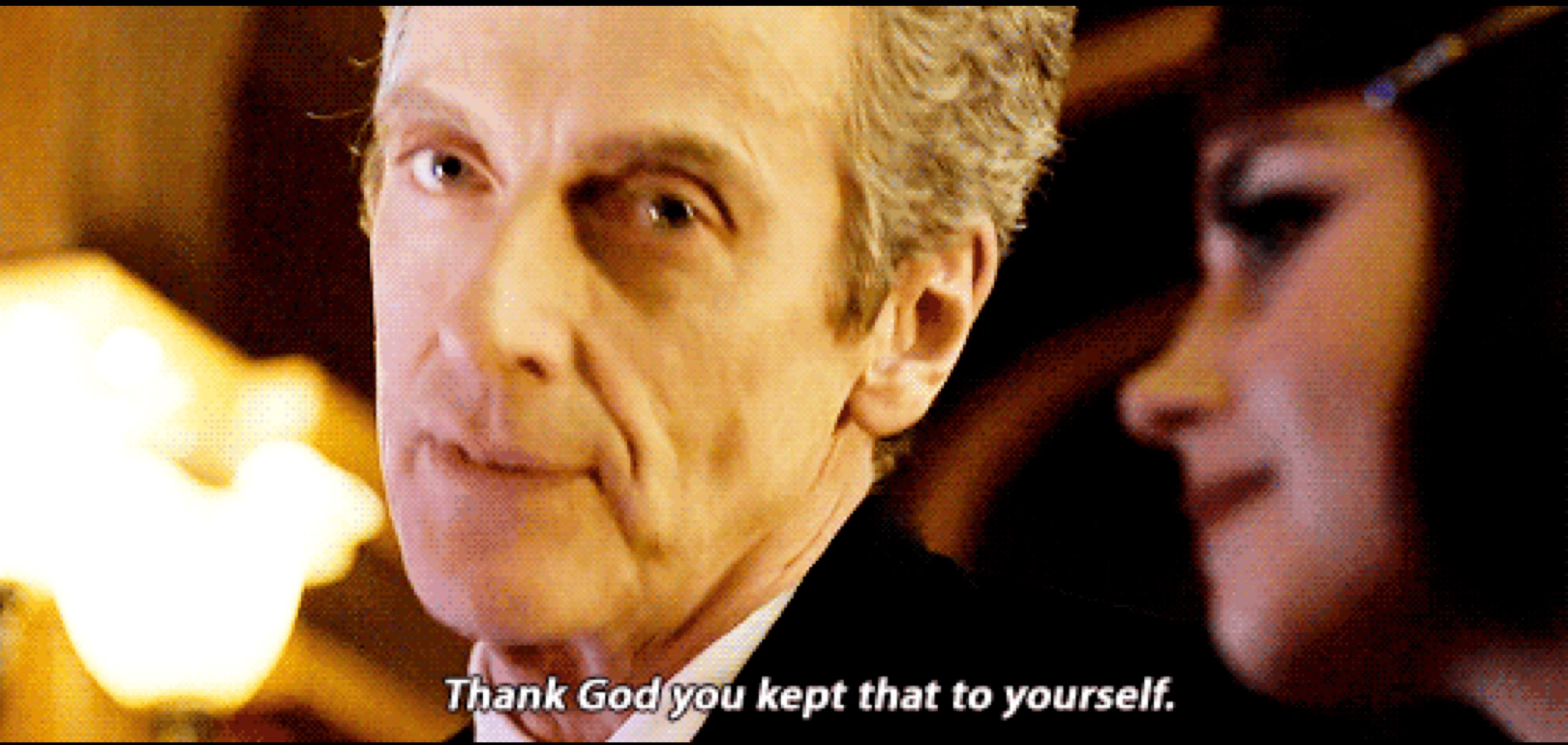
you install from npm  
thoughtlessly  
& this is awesome



**Lesson: stable platforms  
are where ecosystems flourish**

**we did this with a tiny team  
12 engineers now; half that to start**

I had no idea when I started  
that I knew how to do this



*Thank God you kept that to yourself.*

**Lesson: I probably didn't.  
But I could learn.**

**you can too**

**record-scratch noise again**

**I've learned more in the last 3 years  
than in my entire previous career**

**you ready? here comes the drop**

**Lesson: automate everything**

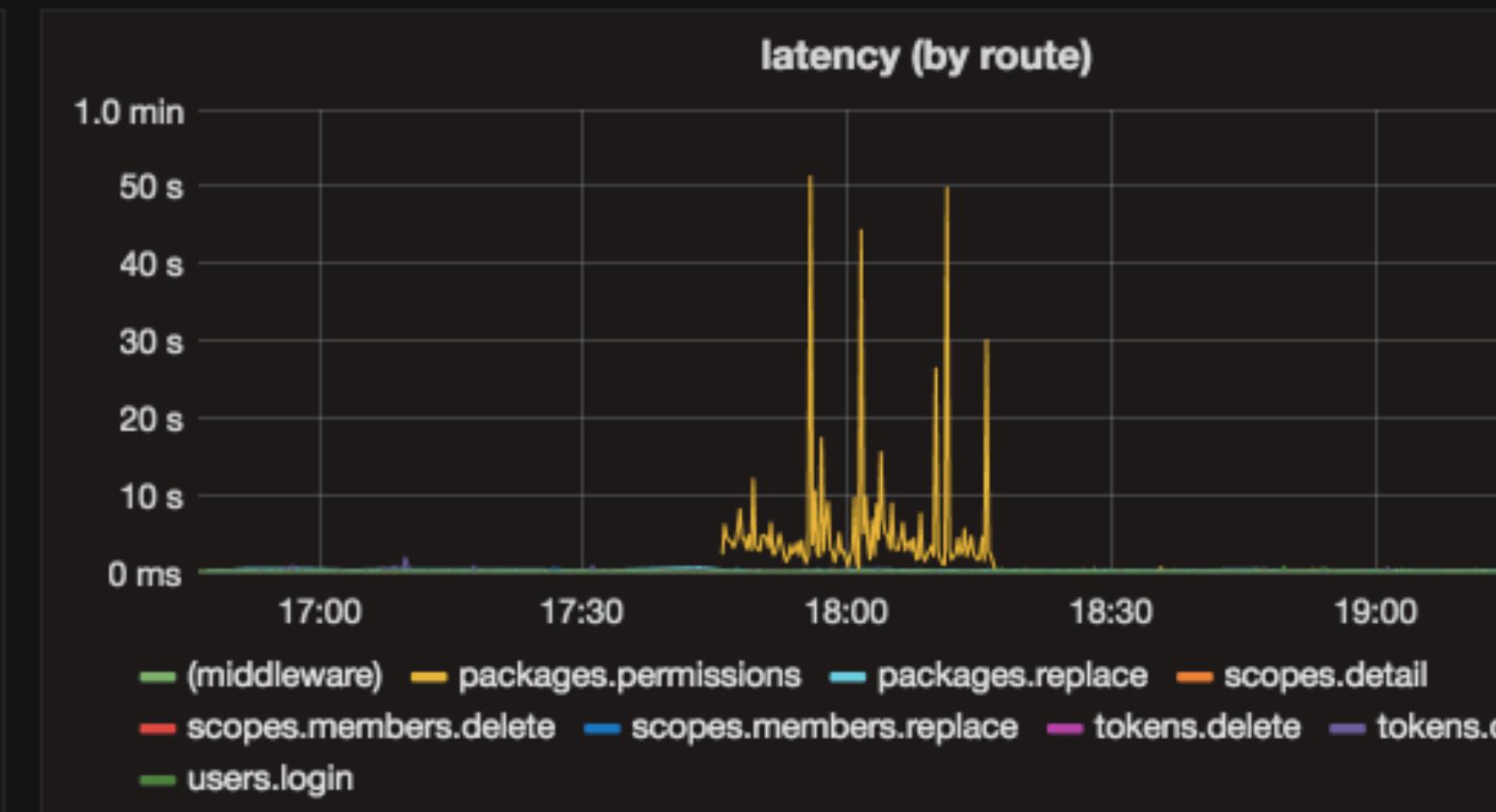
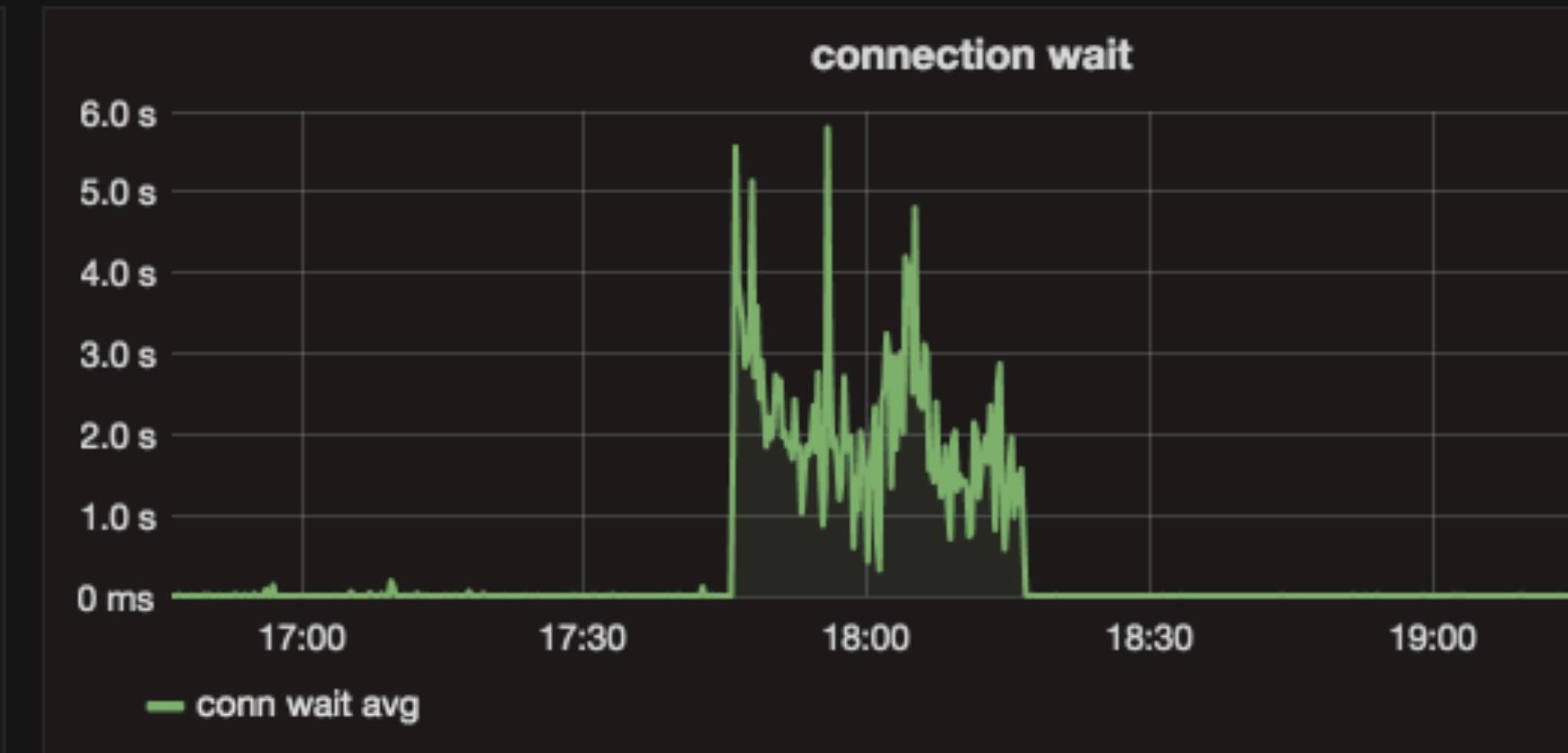
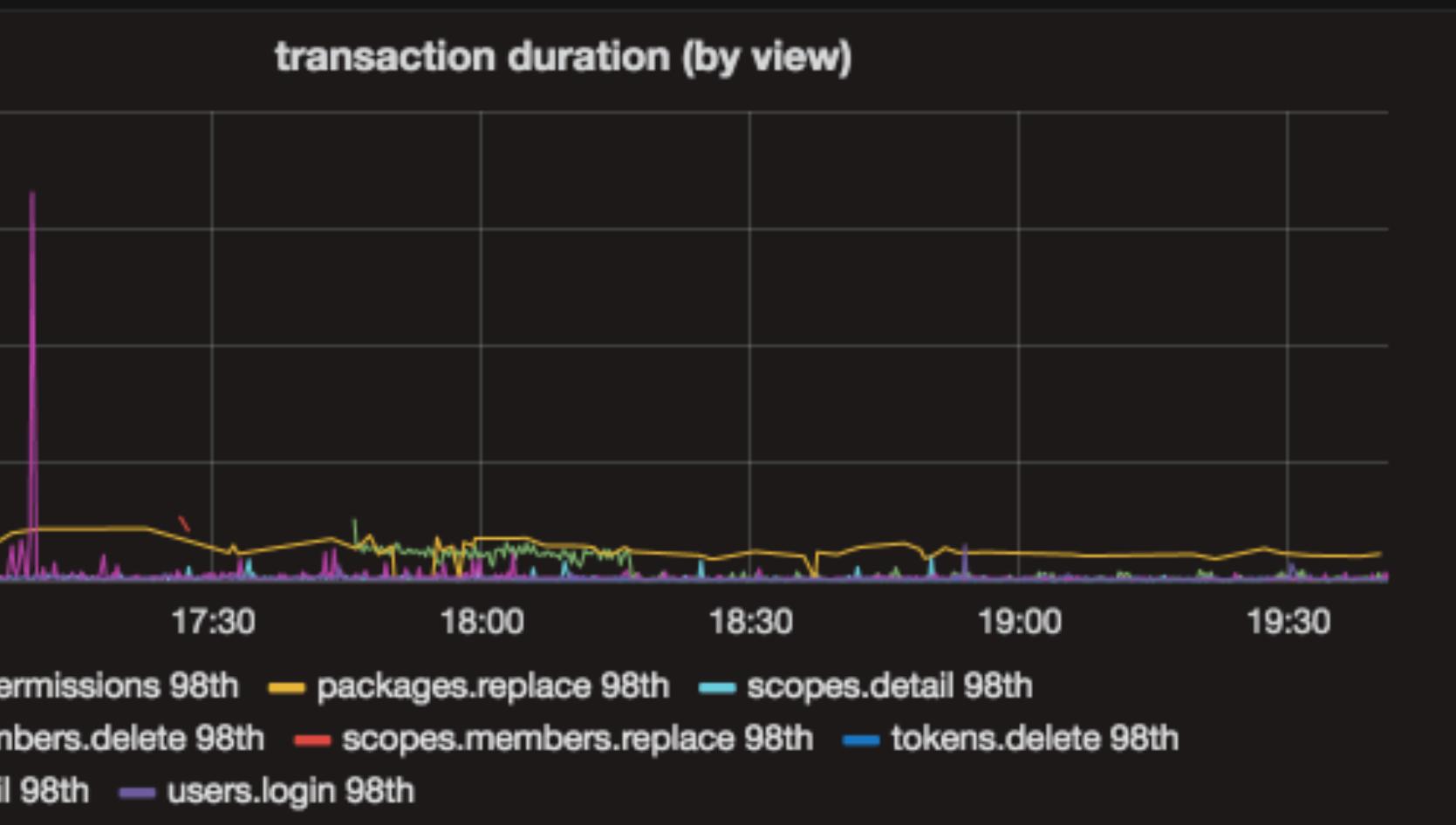
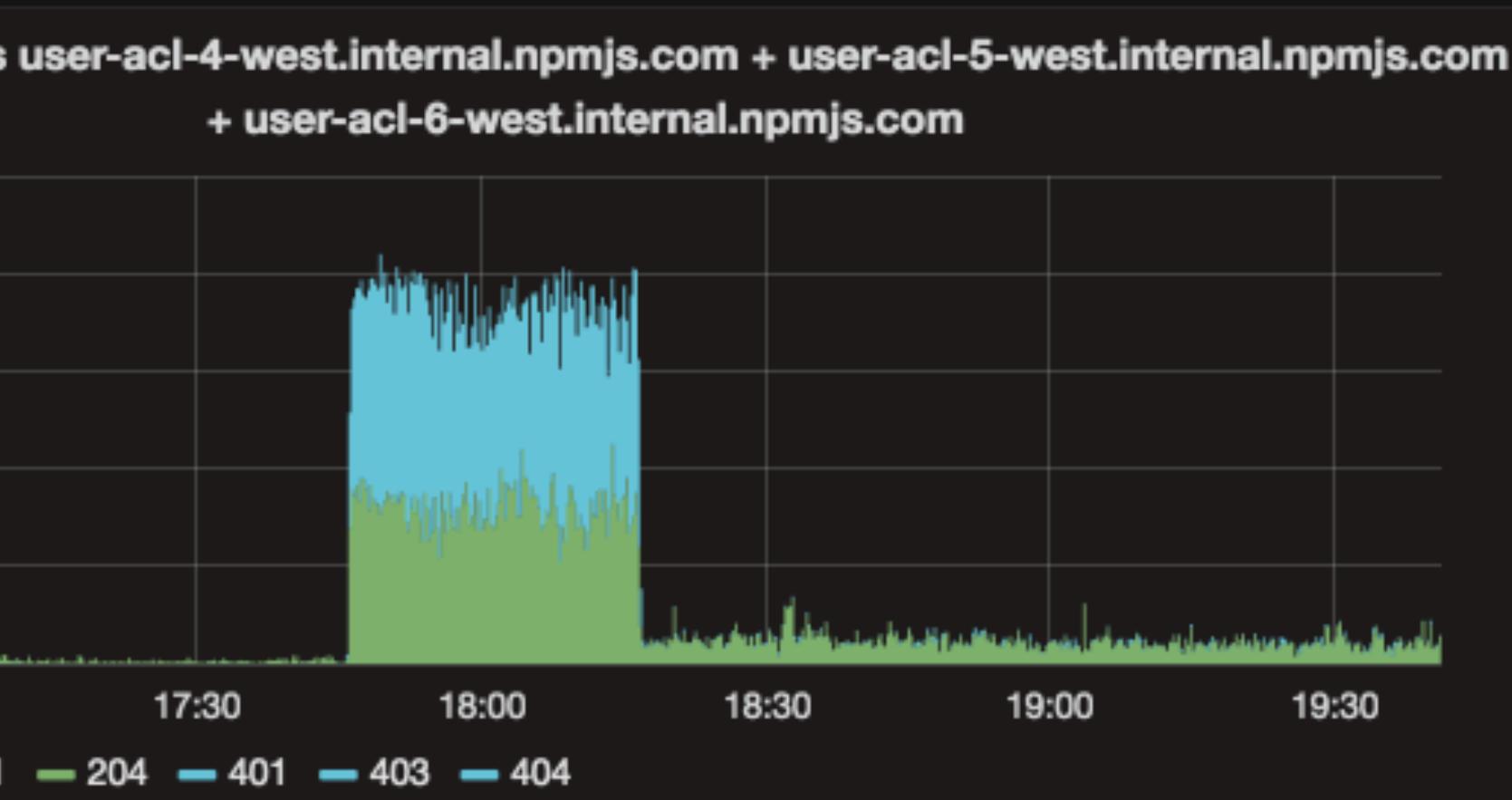
**you're a programmer, right?  
automating things is what we do!**

**automate it so you don't have to  
think about it any more**

**Lesson: observability is crucial**

**if you can't observe your software  
you have no idea what it's doing**

put a metric on it



want to steal our metrics system?  
**[github.com/numbat-metrics](https://github.com/numbat-metrics)**

**Lesson: measure & iterate ftw**



**Cowperthwait** @cowperthwait



Buried on page 14 of our responses to a customer's security questionnaire,  
this candidate for Understatement of the Millennium, cc: [@ceejbot](#)  
[pic.twitter.com/XoF5sAp3p0](https://pic.twitter.com/XoF5sAp3p0)

Oakland, CA

npm, Inc.'s Chief Technical Officer CJ Silverio, and the  
npm culture as a whole, are obsessed with improving  
process surrounding deploying service infrastructure.

**try new things incrementally**

ship a small thing  
that improves your situation

**measure & observe  
improve it incrementally**

**repeat until victory**

**here's a concrete example:  
npm's current API is a legacy mess**

want to replace it all with  
**RESTful APIs** for registry data

a search API and  
our first **React** app in front of it

[find packages](#)

Greetings, seldo



# Hello world!

This is the front page of the search service. It doesn't really need to be here.

**minimal change, measure & iterate  
one little slice of the problem**

**if this works, we'll do it again  
repeat until victory**

**Lesson: simple is good.**

**if you want to be fast  
do as little work as possible**

**make data immutable  
get out of a db & onto a filesystem**

**Lesson: devops is hard  
(but you gotta do it)**

the devops **skillset** is huge

**the devops mindset is balanced**

**development is optimistic  
operations are conservative**

**dev: computers woohoo!**

**ops: computers suck.**

**we only run node LTS releases**

**we only upgrade when we want  
a new JS language feature**

**Lesson: don't get **dogmatic** about tech**

**performance work ongoing**  
**because y'all show no signs of slowing down**

**we're experimenting with Rust  
for our perf hot spots**

**because we love javascript  
but it isn't always the right choice**

**minimal change, measure & iterate**  
**test Rust with one μservice**

**Lesson: all databases are terrible**

**you can only love a database  
you've never put into production**

corollary: your tools  
are always **terriusefulbad**

**now for some harder problems...**

**the non-technical problems**

**Lesson: my job is only sometimes  
solving hard problems**

**my job: identify the right problem  
& hand it to the right person**

**Lesson: people are  
harder than technology**



**ceej season**

@ceejbot

Code is never the challenge. Well-rested  
comfortable people who feel emotionally safe  
have solved every problem I've put in front of  
them.

---

RETWEETS

**754**

LIKES

**1,188**



7:28 AM - 5 Aug 2016

**the **real** challenge is keeping people  
in a place where they can do their best**

I'm **selfish**. I build the engineering  
organization I want to work in.

**collaborative, constructive, curious**

I won't know how I did until  
I can look back in 5 years

node is **still** the most fun  
I've had writing code

I hate it, of course,  
because we always hate our tools

**but node is how I write everything  
from twitter bots to deployment systems**

I want you to type **npm install**  
& have packages flow like **spice**



**Bye!**

npm ❤ you