You’ re reading Signal v. Noise, a publication about the web by Basecamp since 1999. Happy Thursday!

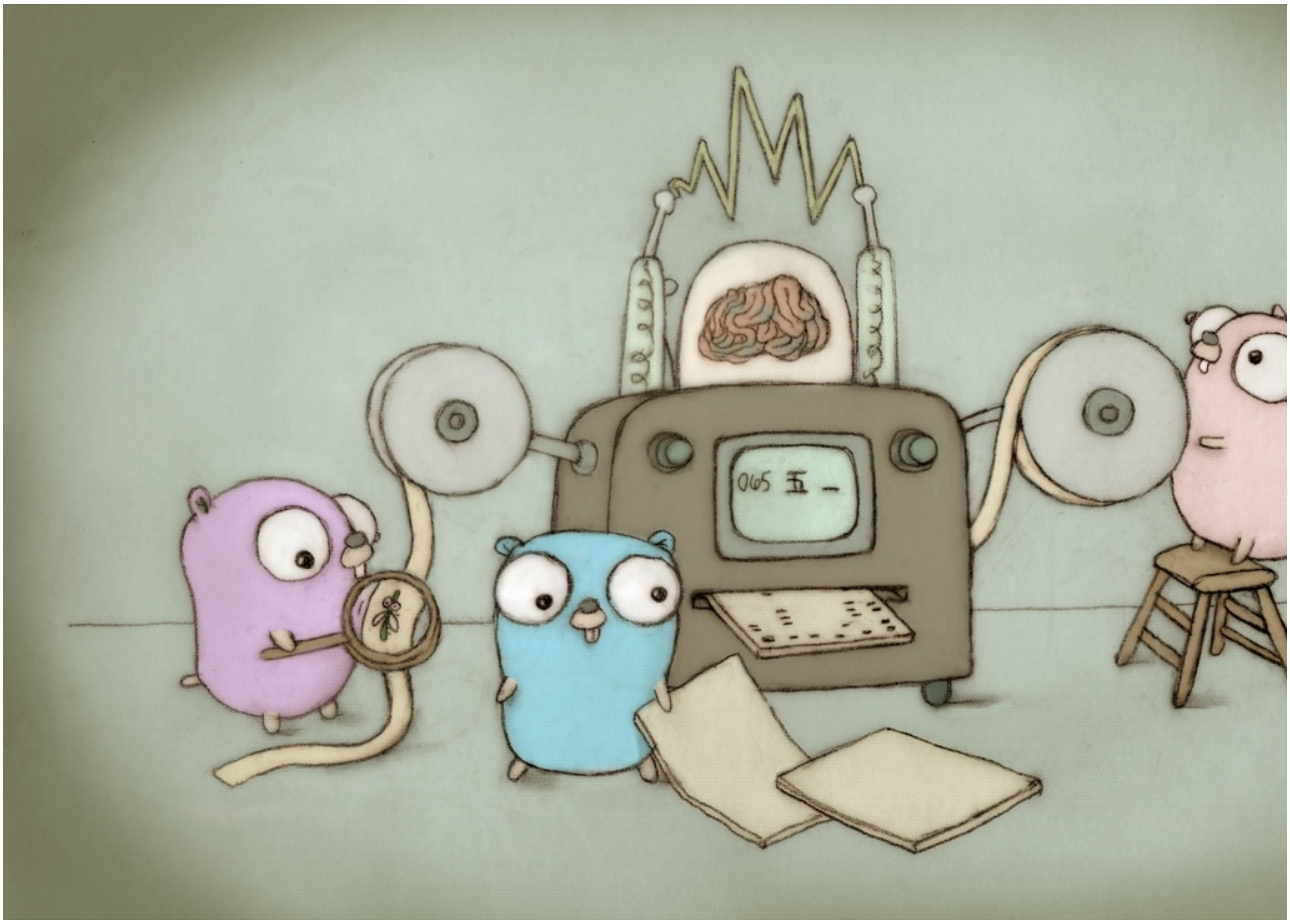
Go at Basecamp

Noah *wrote this on* Aug 31 8 comments

- Tweet (423)
- Share on Facebook (287)
- Share on LinkedIn (24)

Basecamp is a Ruby company. All of our customer facing applications are written with Ruby on Rails, we use Ruby for our systems automation via Chef, we deploy via Ruby through Capistrano, and underneath most rocks you’ ll find a Ruby script that accomplishes some task.

Increasingly, however, Go has found its way into our backend services and infrastructure in a variety of ways:



- Our timeseries data acquisition and storage daemon was rewritten from Ruby to Go in January 2013.
- Our Ruby build scripts build new Ruby packages for our servers via Docker.
- Our log parsing and storage pipeline writes to Kafka, HDFS, and HBase via an assemblage of Go programs.
- We backup our DNS records from Dynect with a tool written in Go.
- We run a multi-master Nagios installation via a Go-based passive check bridge and multi-host notifier.
- We keep our GitHub post-commit hooks in shape using a Go program.
- The server side of our real user monitoring and pageview tracking systems are entirely written in Go.
- We regularly download, decrypt, and test the integrity of our offsite database backups with a Go program.

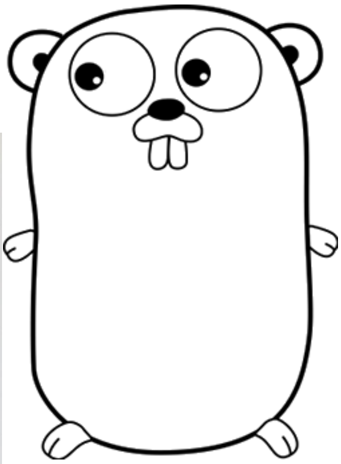
There are also numerous experiments in Go that haven’ t made it into production: keeping multiple memcached instances in sync from packet captures, serving Campfire over websockets, packaging our Rails apps into Docker containers, and more. We’ re also heavy users of some third party Go applications (etcd and sentinel) which power our failover process between datacenters.

Our use of Go is entirely organic. We never sat down one day and decided to start using it; people just started writing new things in Go.

Personally, I like Go because the semantics of channels and goroutines are a great fit for building data pipelines, and the innate performance of Go programs means I don’ t have to think as much about the load that a parser might be adding to a server. As a language, it’ s a pleasure to write in—simple syntax, great standard library, easy to refactor. I asked a few other people why they enjoy working in Go:
Will: *“Go feels perfect for Ops work. The error handling seems to fit so naturally into the way I want to write systems software, and on top of that it’ s good (and getting better) at using multiple cores effectively. Deployment is really simple too, where I’ d have to think about how to package up deps and configure Ruby versions I can now just push an updated binary.”*

Taylor: *“When you are learning a new programming language sometimes you reach a point where trying to solve a real problem furthers your understanding of the language and it’ s strengths. Go’ s fantastic documentation, ease of testing and deployment (compile once and run anywhere via a single binary) are enough to help even a novice write a performant and reliable program from the start. Where you might spend hours debugging a threading bug in your Ruby program you can spend minutes implementing go channels that seem to just work. For even the basic script that needs high concurrency this is a huge win.”*

It’s unlikely that you’ll ever see a fully Go-powered version of Basecamp, but Go has certainly found its way deep into our infrastructure and isn’t likely to go anywhere soon. If you’ve never tried it out, give it a shot today!



Gopher drawings by Renee French and licensed under the Creative Commons 3.0 Attributions license.

Noah wrote *this on* Aug 31 *There are* 8 comments.

Vito Botta

on 31 Aug 15

‘Go on Rails’ sounds nice…

Michael

on 31 Aug 15

I think this is a sensible progression. You should be commended for avoiding trying to do systems work with Node before using Go, which is too often the mistake made when people are looking to ‘upgrade’ from Ruby. Having deep expertise in a language and library ecosystem lets you skip those half-baked bridge technologies in favor of one, relatively painless integration of a mature technology.

Nathan Youngman

on 31 Aug 15

Thanks for sharing. I hope you don’ t mind that I’ ve linked this article from the Go wiki at:
https://github.com/golang/go/wiki/SuccessStories

Doug

on 31 Aug 15

@Basecamp

Would you ever consider replacing your Rails infrastructure with Phoenix?

http://www.phoenixframework.org

Phoenix web framework gives you all of the benefits that Go provides (via Erlang instead) and the web framework is heavily inspired from Rails.

NL

on 31 Aug 15

@Doug – we aren’ t looking for a replacement for our Rails infrastructure, and that’ s not what Go has done for us. With one or two exceptions, we’ ve used Go when writing new stuff—tools that we didn’ t previously have, not to replace something that already existed in Ruby or Rails.

Verónica

on 31 Aug 15

So… months ago I was about to apply for the Basecamp Android team, and decided not to do so because I wanted to do more Go things professionally. If I knew you were using Go too, I would have applied immediately lol. /me weeps.

Ed

on 01 Sep 15

In the future may be Basecamp could take a look at Crystal ? http://crystal-lang.org/

Will Jessop

on 01 Sep 15

@Ed: It depends on a number of things. Popularity (which equates to experience and shared learning), maturity, standard library quality and scope, performance, and of course wether any of us actually are interested in learning it.

In short, maybe one day :)

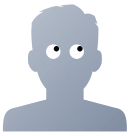
Post *your* thoughts

Name Name (required) Email your@email.com Comment Type your thoughts here…

Basic HTML is allowed: `a` `href` `strong` `em` `blockquote`

We’ d rather not moderate, but off-topic, blatantly inflammatory, inappropriate or vapid comments may be removed. Repeat offenders will be banned. **Let’ s add value.** Thank you.

Post this comment or preview it



About Noah

Noah Lorang is the data analyst for Basecamp. He writes about instrumentation, business intelligence, A/B testing, and more.

Read all of Noah’ s posts, and follow Noah on Twitter.

If you liked this post by Noah, you’ ll probably like reading Ask us anything Thursday at 12:30 p.m. Eastern, Forecasting support response times with the Support Simulator 4000, and API design for humans

New here? Start with our best hits:

- How do you turn inspiration into skill?
- There's always time to launch your dream
- Acquire taste
- Forget the resume, kill on the cover letter
- Four letter words
- … all of “The Best of Signal v. Noise”

Back again? Catch up on our latest posts:

- Growing our audience for The Distance
- Go at Basecamp
- Something's coming 6
- Can old world be more modern than new school?
- Extra Drawings for The Distance
- … all posts on Signal v. Noise

Search for articles or writers

Search Signal v. Noise…

Explore 15 years of Signal v. Noise

- Business
- Design
- Programming
- Support
- Sysadmin
- Writing
- Best of
- Everything

Meet the writers of Signal v. Noise. If you like what you’ re reading, check out our best-selling books REWORK and REMOTE.

Subscribe to our RSS.

Enjoy your Thursday.