Adam McCullough

Experienced Backend/Infra engineer, passionate about agility and reliability through types, tests, and observability.

TheWizardTower@protonmail.com
github.com/TheWizardTower
linkedin.com/in/adammccullough
thewizardtower.github.io

WORK EXPERIENCE

AUGUST-SEPTEMBER 2025 (FT)

Wander
Senior Product Engineer

Role Infrastructure and Full Stack Engineer on a skunkworks team to design and implement the next generation of infrastructure, supporting both rapid local developer experience and bulletproof prod infra, as well as pivoting the codebase to the Functional Programming paradigm through Effect.

Buildkit Stood up a proof-of-concept instance of BuildKit in a Google Kubernetes Engine (GKE) cluster, using SmallStep for certificate generation, external-dns to manage Google's Cloud DNS offering for Ingress endpoints, and mTLS verification (also using certs generated by SmallStep and stored in a Kubernetes secret) on incoming build requests.

Effect Quickly got spun up on and helped to stand up an example service in the Effect library in a local Colima VM that connected to and operated on a CNPG database in the Kubernetes Cluster.

OCTOBER 2024 - AUGUST 2025 (FT)

Capital One (Contracted through Kunai)
Enterprise Data Engineer

Role Terraform expert for a Databricks infrastructure team serving a number of internal customers.

Project - New Databricks Account Unblocked a Skunk-Works project to create a new Databricks account and associated workspaces that integrated with high-touch, which was a top-line strategic goal for the entire corporation. This required troubleshooting and revising Terraform code, creating separate AWS resources (IAM roles, S3 buckets, KMS keys, etc) to create a permission airgap between the new workspace and old ones.

Project - Plan Cleanup Proposed, Implemented, and Deployed several fixes to workspace configurations that resolved non-convergent plans. This significantly reduced the complexity of reviewing Terraform plans for routine changes and updates.

Training - Terraform Led several training sessions across teams on the mechanics of Terraform, how it works, how to reason about your configuration and plan output, and sharp edges in Terraform to look out for.

JULY 2022 - OCTOBER 2023 (FT)

FP Complete
Senior Software Engineer

Role Served as an expert in both robust infra and maintainable, scalable code, available to help customers that had secured a contract.

Project - Server Overhaul Spearheaded a core server rewrite for a customer that suffered from long build times, brittle code, and flaky tests.

- Consolidated codebase from a hybrid Yesod/Servant approach to just Servant to ease maintenance.
- Reduced build time and improved test coverage
- Simplified, commented, and typed extensive parts of the code base
- Published servant-combinators to extend Servant's capabilities

TOOLS

Haskell, Rust, Python, Bash, Terraform, Postgres, Redis, Kubernetes

SPEAKING ENGAGEMENTS

Killed by 'Worse is Better'

LambdaConf 2025

FP has failed to capture the industry zeitgeist, while its effectiveness has only grown. Cover the last time academia lost to 'pragmatic' priorities by examining Lisp and Unix, including the attempts to commercialize both Lisp and the hacker spirit. Closed with a historical anecdote of parallel difficulties of proven-effective methodologies struggling to find adoption in the medical field, and closed with some recommendations/hopes for changes in behavior in academia, industry, and individual practitioners.

Type Safety or Security Theater: Using Servant and Yesod as Case Studies

LambdaConf 2024

Discussed the objectives that are latent in type safety, correctness, and how they help engineering teams achieve business goals. Used various methods of traffic management as a metaphor for cost and effectiveness. Illustrated this by demonstrating that many functions in Haskell are partial, and use error liberally, even when added machinery like ExceptT is present. Then compared and contrasted how Servant and Yesod solve similar problems in endpoint handlers. Concluded with a discussion about what it means to not just write code, but be a steward of it.

Rust's Borrow Checker Proven Correct

LambdaConf 2019

Gave an overview of bugs in various compilers, and how difficult they are to identify and debug. Describe how the Rust devs used formal verification to prove that the semantics of the borrow-checker would produce programs without race conditions. Conclude with some motivating examples to demonstrate why various designs fail the typechecker, and what race conditions they prevent.

- Gave an internal presentation to engineering team about the ergonomics and trade-offs of the two libraries

Project - Rust Conversion Wrote a utility to convert a Haskell server binary to a collection of flat files, enabling us to host the same content on Cloudflare, rather than on an AWS instance.

AUGUST 2020 - JUNE 2022 (FT)

Daisee
Lead Systems Engineer

Role Served as Head of Operations for a SaaS company that was just graduating from start-up to scale-up

Project - Infra Overhaul Spent significant time debugging and overhauling the existing legacy terraform code

Project - App Re-Design Lead the infra-side of an app re-design

- Migrated to an on-demand instance approach managed by ECS, which significantly reduced costs without affecting throughput
- Consolidated to a multi-tenant approach, so adding new customers did not require an infra change

Project - Haskell Work Migrated prod code to a new program monad in core-program

- Included many optimizations to run-time configuration and developer conveniences for things like logging or command line argument parsing

Project - Observability Added observability to Haskell code via Honeycomb and a purpose-built Haskell OpenTelemetry library.

- This helped enormously in both finding fruitful places for optimization work, as well as quickly understanding production outages

MARCH 2018 - JULY 2019 (FT)

Facebook
Production Engineer

Role Worked as a Production Engineer (PE) on the Feed and Stories Team (FAST)

- FAST was responsible for News Feed, the centerpiece of the Facebook web experience, both w/r/t MAU and ad revenue.
- FAST was a large team (1.5k at intern season), while the embedded PE team was much smaller (7)
- We oversaw server infra, compute capacity planning, building and monitoring new releases of the software that ran news feed, aggregator

Oncall Member of the push oncall rotation for Aggregator

- identify regressions in engagement, ad revenue, or new crashes arising from race conditions or Thrift configuration mismatches

Project - ASAN Added per-diff Address Sanitization (ASAN) canary tests for aggregator

- Required working cross-functionally with Sandcastle (CI), Phabricator (code review + test stewardship), as well as securing server capacity, and deploying into production use

Stewardship Also spent time refactoring legacy C++ projects to use safer, more modern idioms, which lead to updating and improving code for the iterators of some data structures in the Folly library.

NOVEMBER 2016 - SEPTEMBER 2017 (FT)

IMVU
Software Engineer I

SPEAKING ENGAGEMENTS, CONT'D

Monad Transformers for the Easily Confused

LambdaConf 2018

Begin with a review of Monoid, Functor, Applicative, and Monad typeclasses. Demonstrate that Monads do not compose, then demonstrate how Monad Transformers solve this problem, first with the IdentityT monad, then the MaybeT monad.

Lambda Calculus for the Easily Confused

LambdaConf 2017

Walkthrough of the mechanics of Lambda Calculus, with a particular emphasis on how these mechanics can give an intuition for things in Haskell, like partial function application or higher-kinded types. Concludes with a demonstration of the Y combinator.

VOLUNTEER WORK

Board Member, IT Director, People's Pantry of Ferry County

Organized and ran a Haskell class at IMVU, 2016

President, ASULUG, August 2010-August 2012

PERSONAL ACHIEVEMENTS

Amateur Extra Radio License

Callsign AG7YC

Stackage Library

servant-combinators, which adds some primitives to enable Servant users to access core components of the inner WAI request.

SKILLS

Thorough
Public Speaker
Customer Focused
Experienced Mentor
Functional Programming
Systems Analysis
Linux Administration
Robust Systems Architecture and Design

Role Worked as a backend engineer on the FIRE team

- FIRE was responsible for finding and fixing problems in a diverse legacy codebase, including PHP, Haskell, JS, and CSS.

Project Maintained and extended a customer marketing tool

- Tool extracted data about customers, such as if they'd spent money, when they joined, if they had a "special someone", etc.
- This information was used to tailor promotions for customers.

Project Designed, implemented, and deployed a backup-and-restore system for Scylla

- Scylla is a C++ re-implementation of Cassandra
- This required extensive reverse-engineering, as documentation for the backup procedure was extremely thin at the time.
- This also managed and pruned backup files according to a retention policy, to help with data storage volume constraints

JUNE 2013 - NOVEMBER 2016 (FT)

IMVU
Systems Engineer II

Role

- Maintain, triage, and resolve issues impacting the production cluster or service quality, 24/7
- Root-cause incidents, identify steps to prevent it repeating in the future, then implement those steps.

Oncall Carry a pager for issues that affected service quality, 24/7

- Responsibilities include responding to pages within 15 minutes while oncall, and communicating to the company: observed impact, path to remediation, and time to resolution.

Project Replaced Varnish caches with Apache Traffic Server

- Varnish had a significant memory leak on the kernel version deployed to prod. Switching to ATS significantly reduced oncall load and improved service quality.

EDUCATION

2007 - 2013 **Computer Systems Engineering**
Ira A. Fulton School of Engineering
Arizona State University