Arthur Axel fREW Schmidt

frew.co, resume@frew.co, (228) 369-4492, github.com/frioux, Santa Monica, CA

I solve problems, both with software and a thoughtful approach.

Major technologies:

- Go (both web services and systems automation)
- Terraform
- AWS (EC2, S3, SQS, DynamoDB, SecretsManager, Athena, Aurora)
- Kubernetes
- Kafka

Experience

Senior Staff Software Engineer @ ZipRecruiter

Jan 2021 → Present

`go data-lake dynamodb kafka sqs cloudflare nginx scala spark delta-lake`

Projects I've been involved in as Senior Staff Software Engineer include:

- Critical http service: helped shepherd a long overdue confirmed opt-in project leveraging Go, DynamoDB, SQS, and Kafka
- Data lake system: designed and helped implement a system to store job seeker resume data in a data lake accessible via Athena or Spark. Uses S3, Delta Lake, Scala, Go, Kafka.
- Data lake system: designed and helped implement a system to detect bot interactions using NGINX, CloudFlare, Scala, Spark, Delta Lake, and Kafka.
- RDBMS account management: designed system to manage accounts and related secrets, in all tiers. Helped an engineer on another team implement this.
- Helped maintain a cross team weekly happy hours despite over a year of pandemic exile.

Staff Software Engineer @ ZipRecruiter

Jul 2018 → Jan 2021 (2 years, 7 months)

`go apache-kafka kubernetes amazon-web-services prometheus`

Projects I've been involved in as Staff Software Engineer include:

- Critical http service: Mixer logs all impressions, runs cross search engine A/B Tests, is written in Go but safely modifiable by even engineers who are not well versed in Go. Uses DynamoDB.
- Kubernetes related infrastructure - Secrets management with AWS SecretsManager, prometheus libraries, Configuration shipping, manifest modification and validation, public ingress
- Kubernetes Migration Strategy - ZR Factors as a more prescriptive version of 12 Factors, Dana as a transitional container management tool
- Revamped logging pipeline - involving filebeat, kafka, kafka connect
- Best Practices for Go use within ZipRecruiter

Senior Software Engineer @ ZipRecruiter

Jul 2015 → Jul 2018 (3 years, 1 month)

`perl mysql catalyst dbix-class amazon-web-services`

As one of the members of the core team I was responsible for the infrastructure that runs ZipRecruiter.

Here are some blog posts I've written about cool stuff I've done at work:

- AWS IAM usage at ZipRecruiter

- How to bolt timeouts on to MySQL 5.6
- How to bolt timeouts on to an internal webservice
- How to efficiently log or count database queries
- Email Bot for explaining AWS Retirement Notifications

Other things I've done which I have yet to write much about include:

- Across the board logging and monitoring improvements
- Memory reduction in web workers
- Reduction in load on the central database
- Performance improvements in realtime code

Other Work History

- Lead Architect and Technical Manager @ Micro Technology Services Incorporated; Oct 2010 → Jul 2015
- Web Developer @ Micro Technology Services Incorporated; Aug 2008 → Oct 2010
- Programmer @ Loma de Luz; May 2007 → Aug 2007
- Programmer @ Sterling Commerce; May 2006 → Aug 2006

Education

B.S. Computer Science and Math; LeTourneau University; 2004 → 2008

Open Source

- DBIx::Class::Candy, Started Jul 2010 - Sugar for the Perl ORM; Creator and Maintainer
- DBIx::Class::DeploymentHandler; Started May 2010 - Best in class database deployment tool that leverages DBIx::Class as well as easing the deployment of DBIx::Class schemata; Creator and Maintainer.
- DBIx::Class::Helpers; Started Oct 2009 - Significant additions to DBIx::Class (the Perl ORM) via a large number of plugins; Creator and Maintainer (though there are a handful that have been contributed.)

Public Artifacts

- Mixer Post Mortem: deep dive into a total outage
- A Custom Supervisor to Solve Weird Problems: A blog post about the crazy things you sometimes have to do to solve a problem.
- Adding Autoreload to `srv`: tricky code in some of my OSS showing how to automatically reload a web page when a file on disk changes
- `context` Deadlines in Go: discussion of how timeouts (and similar errors) are handled in Go
- Go Subtest Tips: tips and tricks for writing subtests in Go
- Writing a Go Linter: how and why I wrote a custom linter for Go code.
- Testing Perl Clients and Go Servers: A triumphant post describing how I was able to test client and server code together without mocking either.