

# Michael Dresser

michaeldresser.io | michaelmdresser@gmail.com

## EXPERIENCE

### STACKWATCH (KUBECOST) | SENIOR SOFTWARE ENGINEER

Feb 2022 - Present | Remote

- Designed and implemented a filtering language for core APIs. Built a lexer, parser, and default interpreter for the language, enabling previously-impossible queries for users. (Go)
- Owned a company goal: K8s-native cost-aware workload scaling. Interviewed users, support, sales, and stakeholders; produced feature concepts; and drafted a foundational plan for a new product area.
- Maintained and extended a K8s controller and CRD which provided simple interfaces for complex cost-optimization procedures. (Go)
- Lead design discussions about schema migrations, driving consensus; followed with implementation for backwards-compatibility with all previous data formats. (Go)
- Onboarded, mentored, and supported multiple junior engineers.
- Demoed Kubecost in front of Kubernetes community groups.

### STACKWATCH (KUBECOST) | FULL STACK ENGINEER

Jan 2021 - Feb 2022 | Remote

- Designed, built, and marketed an open-source CLI frontend. (Go)
- Improved performance of savings recommendations by orders of magnitude, leveraging statistics and interval algorithms. (Go)
- Designed, built, and marketed a feature to automatically determine if Kubernetes workloads could be rescheduled onto spot nodes, leading to observed savings of >50%. (Go, TypeScript, React)
- Proposed and built a benchmarking tool for evaluation of changes on simulated data. Reduced feedback loop of benchmarking at real-world scales from 15+ minutes to three minutes. (Go)
- Designed and built a nightly CI+release process in GitHub Actions, enabling users to test bleeding-edge features with ease.
- Wrote technical blog content to share features and K8s knowledge.

### GOOGLE | SOFTWARE ENGINEERING INTERN

May 2020 - Aug 2020 | Remote

- Contributed to Kubernetes test infra, simplifying end-to-end test workflow in an environment running >90,000 tests/month.
- Contributed patches to repositories across the Kubernetes project.

### CU BOULDER | GRADUATE STUDENT STAFF

Aug 2019 - Dec 2019 | Boulder, CO

- Held office hours and graded for Linux Systems Administration.

### TWILIO | SOFTWARE ENGINEERING INTERN

May 2019 - Aug 2019 | Denver, CO

- Designed+built a microservice to enable testing. Moved service into Kubernetes. Drove adoption in 3 teams.
- Created Go service and automatic pipeline to recover failed operations by recirculating Kafka messages.

### LOGRHYTHM | SOFTWARE ENGINEERING INTERN

Jun 2018 - Aug 2018 | Boulder, CO

- Developed Go microservice to write logs to GCP Pub/Sub.
- Developed Go+Elasticsearch microservice for preventing PII cross-contamination.

## EDUCATION

### UNIVERSITY OF COLORADO BOULDER

B.S. AND M.S. IN COMPUTER SCIENCE  
GPA: 4.0 / 4.0

## SKILLS

### LANGUAGES

Go • Python • TypeScript • Javascript • Clojure • Bash • Lisp • SQL • Scala • OCaml • Nim • Rust • Java

### TECHNOLOGIES

Linux • Kubernetes • Docker • Git • SQLite • Prometheus • GCP • AWS • Jekyll • REST • Spark • Kafka

## COURSEWORK

### GRADUATE

Algorithms  
Programming Languages  
Machine Learning  
Linear Programming  
Data Mining  
Datacenter Scale Computing

### UNDERGRADUATE

Operating Systems  
Big Data Architecture  
Linux Systems Administration

## INDIVIDUAL PROJECTS

### TEEMO TOOL

Clojure, TypeScript, Python, Mithril.js IRC bot, API server, and website that gather and display information about the betting minigame on the Salty Teemo Twitch stream. Includes a crawler for the Riot Games API and Jupyter notebooks to analyze the gathered data.

### THESIS: ARUGULA

Scala  
Arugula is a probabilistic programming language that I designed and implemented for my thesis. It has special semantics that support delayed sampling of distributions combined with symbolic inference where possible, which improves the accuracy of estimators.