Ryan Kemper

Education

2014–2018 **Computer Science**, *UC Santa Barbara - College of Engineering*.
BS

Experience

May 2019-Present **Site Reliability Engineer**, *Invoca*, Santa Barbara.

- Revamped our logging Elasticsearch cluster configuration, taking us from an unstable, over-sharded configuration to a performant and stable daily index model
  - New configuration processed **1.3 billion** documents per day while reducing total cost on the order of tens of thousands of dollars per year
  - Cluster went from experiencing "red cluster status" (data loss) every hour to **no further production** events in over 7 months
- As part of a 24/7 on-call rotation, performed triage and incident response for our Kubernetes/chef managed infrastructure as well as our **ruby-on-rails** application and associated python microservices.
- Lead the response to production incidents while maintaining composure
  - Advocated for our customers, pushing us to proactively notify about impacting events such as dropped phone calls
  - Worked with others to perform **root cause analysis** and blameless post-mortems, identifying and driving improvements to prevent future incidents
- Securely managed production secrets through Hashicorp vault
- Operated and troubleshooted several Kubernetes and Chef-based production environments
- Deployed Falco, a container native runtime security and compliance solution, across all our Kubernetes and Chef-managed nodes

Jun 2018-March 2019 **Cloud Operations Intern**, *Invoca*, Santa Barbara.

- Plumbed cloudwatch Elasticsearch and Logstash metrics and constructed information-dense Grafana dashboards
- Developed and tuned alerts using Graphite-based monitoring stack; writing one-off scripts as necessary to back-test against historical data
- Wrote consistently high quality documentation containing specific runbooks and high level conceptual explanation of critical infrastructure like Elasticsearch
- Merged a patch (extended functionality) to the open-source static code analysis tool Brakeman, working with my manager to optimize Docker caching layers and create a new Dockerfile for easy usage in a CI pipeline => https://github.com/presidentbeef/brakeman/pull/1252
- Wrote a module for onboarding laptop script which idempotently guided the user through setting up a password-protected 4096-bit RSA ssh key to enforce a standard of security excellence
- Wrote a module for developer bash profile which automatically displays current Kubernetes context and namespace to ease common UX difficulties in using Kubernetes

Fall–Winter 2018 **UCSB CS Capstone team member - 1st place**, *LogMeIn (sponsor)*, Goleta.

Developed proprietary software that uses machine learning to offer targeted feedback for public speaking:

- Implemented NLP techniques like TF-IDF to identify keywords
  - Used nltk brown corpus to extract english term frequency and inverse document frequency data
  - Wrote Python scripts to extract thesaurus data and corpus word frequencies;
- Maintained 3 backend modules (text/audio/video):
  - Wrote shell scripts using sed/awk to integrate backend and frontend repositories;
  - Specified and implemented backend API for use in frontend
- Used parallelization to speed up backend, reducing processing time 2-3x:
  - Multithreading to parallelize API calls to Google Cloud Vision and IBM Watson
  - Multiprocessing for cpu-bound tasks like OpenCV haarcascades, ffmpeg mp4 conversion
- Wrote decision tree logic to convert raw analysis results into plain english feedback for user

Primary Languages

Python 3	Advanced	*Preferred language for machine learning, scientific computing*
Ruby	Advanced	*Preferred language for scripting/general computing, metaprogramming*
C	Advanced	*Preferred language for low-level systems*
Java	Intermediate+	*Strong familiarity, but not preferred language. Often read Java source for projects like Apache Lucene, Elasticsearch, etc*
Go	Beginner	*Mainly poking around in Kubernetes / Vault / Terraform source code*

Core Skills

Git	add, commit, push, feature branching, stashing, tagged commits, remotes, reflog, rebase	**Scripting**	regexes, safe file handling, multiprocessing, JSON/csv/other common data output formats
Security	SQLI, CSRF, XSS, privilege escalation, buffer overflows / shellcode injection via env vars	**Cryptography**	secure hash functions (like the sha-2 family), salting, asymmetric key encryption (like gnupg)
Linux	Extensive production experience operating linux instances at scale, primarily Debian/Ubuntu/BusyBox	**Concurrency**	Synchronization primitives such as mutexes (locks), semaphores, wait/join, thread vs process

Public Speaking

(Link) Pitching an idea to college students @ Capstone UCSB