Seth Moore

Senior Site Reliability Engineer
Pittsburgh, PA / +1-412-475-8250 / sethdmoore@gmail.com
sethdmoore.com / github.com/sethdmoore

Professional Summary

Senior Site Reliability Engineer with 13+ years of experience in infrastructure engineering and automation. I deliver **infrastructure as code via Terraform** and maintain resilient **Kubernetes** environments for high-traffic production workloads. Highly proficient in **Python and Golang**, I develop agentic AI workflows, custom CLI tools, and CI/CD pipelines to reduce operational overhead. I **leverage observability** to identify single points of failure, refine alerting rules, and fortify system stability.

Core Technical Skills

- **CI/CD and Pipelines:** Jenkins, Travis CI, Groovy
- **Cloud and Orchestration:** AWS: Kubernetes (EKS), ECS, Fargate, Lambda
- **Databases, KV and Queues:** PostgreSQL, MySQL, MongoDB, Redis, RabbitMQ, Consul
- **Infrastructure as Code:** Terraform (Advanced Module Design), CloudFormation
- **Linux and Shell Mastery:** Alpine/Arch/Debian, POSIX shell, coreutils, vim/nvim, tmux, make
- **Networking:** Unifi, Transit Gateway, IP Overlap (Internal NAT), VPC
- **Observability and Security:** Datadog, Prometheus, Grafana, Graphite, Wiz, SonarQube, Nessus
- **Programming:** Python, Golang, NodeJS

Professional Experience

Senior DevOps Engineer | Proofpoint, Inc *2021 – Present*

- **Infrastructure Automation:** Authors and maintains terraform that rapidly provisions entire production accounts. Delivers a feature complete stack, with peered VPCs, RDS databases and route53 records.
- **Terraform Composability:** Develops and maintains reusable, composable **Terraform modules** used across multiple business units to accelerate project delivery.
- **CI/CD:** Writes and maintains various Groovy pipelines to ship various applications to AWS and EKS.
- **Network Engineering:** Developed an internal NAT solution using **AWS Transit Gateway** to enable connectivity between overlapping CIDR blocks across business units and VPCs.
- **Reliability Engineering:** Improved AWS EKS cluster performance by writing Lambda lifecycle hooks (Python). Resulted in a **100x reduction in 500 errors** during EKS Node scaling events.
- **Secrets Management:** Developed `vaultcli` (Python) to automate the migration of secrets from git repositories to **AWS SSM Parameter Store (SecureString)**.

Senior DevOps Engineer | Nomis Solutions *2017 – 2021*

- **Microservices:** Built `NamesAPI` (Golang) to standardize DNS records, tagging, and Terraform resource naming.
- **Scaling:** Wrote `services-scaler` (Python) to dynamically scale EC2 java worker nodes based on RabbitMQ queue depth.
- **Monitoring:** Overhauled legacy Nagios monitoring into a modern **Datadog** stack.
- **Automation:** Developed `NomisPy` - a composable, org-specific Python library for consistency and automation.

Senior Infrastructure Engineer | Forever, Inc *2015 – 2017*

- **Monitoring:** Developed `datamon` (Python) to ship high resolution Postgres metrics for observability.
- **Observability:** Wrote `qwatch` (Golang) to ship high-resolution AWS SQS metrics to Librato.
- **CI/CD:** Built a cross-platform "`Provisioner`" (Golang) tool to automate package installation and config management runs on Linux and WIndows.

DevOps Engineer | BrandingBrand *2013 – 2015*

- **Orchestration:** Architected fault-tolerant **Mesos/Marathon/Zookeeper** infrastructure.
- **Load Balancing:** Wrote `MISHAP` (Golang) to dynamically update **HAProxy** routing from Redis data, achieving 1s resolution for moving containers.
- **DevEx:** Created `stage-me` CLI tool, allowing developers to spin up full application environments with a single command.

Site Reliability Engineer | Livestream *2012 - 2013*

- **Monitoring:** Wrote (and rewrote) scripts that pushed actionable alerts to Nagios monitoring.
- **Infrastructure Automation:** Wrote puppet manifests to manage infrastructure. Shipped collectd and statsd daemons to the hybrid EC2 / datacenter infrastructure.
- **Observability:** Built detailed Graphite dashboards to trace hard to debug issues at scale.