

Profile

Senior Site Reliability Engineer with ten years experience in various development and operations roles. Background in electrical engineering, embedded software, and IOT. Converted to SRE through necessity rigorous cloud adoption. Five years experience in the cloud native space designing highly-available solutions through GitOps with a CI/CD first approach to IAC. Achieving this using my diverse skillset which includes various languages such as Golang, NodeJS, Python, Bash, and HCL to provision cloud-native applications, the supporting services they require, and the infrastructure itself.

Technical Skills

Languages: Golang, NodeJS, Python, Bash, Java, C#

Markup: YAML, Markdown, JSON

Communication: Slack, Teams, Jira, Confluence

CI/CD: Google Cloud Build, Google Cloud Run, Bitbucket Pipelines, AWS Lambda, Travis, JenkinsX, and CodeFresh but prefer a Kubernetes CRD based approach to defining pipelines, for example Tekton.

Databases:

- Relational: PostgreSQL, MariaDB, MySQL, Amazon Aurora, and CloudSQL
- Document Store: MongoDB, RavenDB
- Time Series: Prometheus, Timescale, InfluxDB

OS: Linux, Windows Server 2012 r2

Clouds: AWS and Google Cloud Platform (GCP) - various offerings in both.

Monitoring & Alerting: Grafana, Uptime Robot, DataDog, PagerDuty

Observability: Honeycomb.io, Jaeger Tracing

Security & Compliance: Dome9, WhiteSource, Prisma, Aqua

Miscellaneous: Docker, RKT, Terraform, Consul, Vault, Git (GitHub and Bitbucket), Kubernetes (kubeadm, kops, k9s, k3s, GKE and EKS), some Chef and Puppet, Redis (Self, Memorystore, and ElastiCache), RabbitMQ, Traefik, NGINX, ha-proxy.

Experience

Sr. Site Reliability Engineer

ViewDo Inc

Denver & Remote

March 2019 - November 2019

Much of my work here revolved around automating operational toil. Highlights include writing applications to handle dynamically creating and provisioning Kubernetes namespaces for feature branches across 11 different repositories, reconcile DNS records against a source of truth and add/remove records as

needed, reconcile our approved IP address list against a source of truth and dynamically update firewall rules based on it's contents, and a complete managed service migration from AWS to GCP.

- Designed, documented, and implemented a GitOps CI/CD pipelines first workflow approach to developing and releasing an existing software stack. (Bitbucket, Cloud Build, Cloud Run, Kubernetes CRD)
- Terraform and some CI/CD magic to provision infrastructure and maintain declared state across multiple GCP and AWS accounts.
- Triage and Postmortem issues and outages and convert those into actionable items and tickets.
- Set up Kubernetes clusters on GKE to host a myriad of microservices for the DXP-Platform
 - Creation and provisioning of dynamic namespaces on a per-branch basis on request.
 - Fully automated Deployment, Provisioning, and Maintaining databases and messaging services as statefulsets in those dynamic namespaces.
 - Autoscaling preemptive node pools for stateless services.
 - Wrote clear, maintainable, extensible CRDs to extend the Kubernetes API to enable Pipelines, PipelineSteps, and PipelineJobs for use as an internal and secure CI/CD solution.
- Configured & Deployed HA RabbitMQ Cluster
 - Configured with Terraform, Vault, and Consul across multiple GCP regions with nodes in multiple zones in each region.
 - Forged a custom machine image with start up scripts to automatically join the cluster.
 - Configured as an autoscaling group. Triggered as messages held in RAM hit various thresholds via Prometheus metrics and a Lambda that triggers a pipeline.
 - Scrape metrics from Prometheus and implemented a Grafana Dashboard to reflect the state of the cluster and provide some alerting.
- Configured & Deployed HA RavenDB Cluster
 - Configured with Docker-Compose, Terraform, Vault, and Consul across three GCP regions, with nodes in two availability zones in each region.
 - Traefik LB with Consul Backend to provide SSL for Admin Console on each node authenticated via Client Certificates.
 - Produced a custom machine image with start up scripts to automatically join the cluster.
 - Integrated Grafana Dashboards to reflect the state of the cluster, databases it holds, and provide some alerting in specific situations, like a node failure, CPU and RAM usage, advanced alerting regarding latency and performance.
 - Designed a pipeline to add and remove nodes from the cluster.
- Developed a command line application in NodeJS that is used in all of our CI/CD pipelines.
 - Capable of triggering other CI/CD pipelines.
 - Enables Semantic Versioning via GitVersion.
 - Pushes container images to a registry.
 - Releases Jira issues.
 - Reconciles our active Kubernetes namespaces against branches across several repositories.
 - Get environment variables from Vault secrets and/or Bitbucket repository variables for use with envsubst. Helps keep code DRY.
 - Reconciles a map containing approved IP addresses for remote access to certain segregated services and updates GCP firewall rules based on the addresses.

- Developed a command line application in Golang which reconciles existing DNS records with currently available hosts registered in load balancers then automatically updates records as needed.
 - **Technical Environment:** GKE, Bitbucket Pipelines, Cloud Run, Cloud Build, GCR, Docker, Memorystore, Cloud SQL, RavenDB, RabbitMQ, NodeJS, Golang, Bash, C#, Terraform, Vault, Consul, Traefik.
-

Site Reliability Engineer Influence Technologies Inc

Denver & Remote

October 2018 - February 2019

In this role I facilitated migrating Rackspace based Windows VMs to AWS EKS and other AWS managed services such as ElastiCache, S3, and RDS.

- Managed, Implemented, and Achieved the GDPR Compliance Plan
 - POC for a pure pipeline approach to migrating Rackspace based production VM infrastructure to EKS.
 - Helped containerize and break apart .Net Core microservices.
 - Wrote a pipeline to provision the EKS cluster.
 - Migrate MySQL databases to Aurora DB in RDS.
 - Configured CDNs for static assets.
 - Configured S3 bucket lifecycle policies.
 - Configured the VPC and all IAM rules following the principle of least privilege.
 - Terraform with an S3 backend for declared state.
 - Combination of CI/CD tools to facilitate the migration of the main platform applications to EKS using a canary deployment strategy over the course of a few days.
 - POC - Cloudformation vs Terraform.
 - **Technical Environment:** Rackspace, AWS, Windows Server 2012 r2, Amazon Linux, ServerDensity, Prometheus, Grafana, Dome9, Octopus Deploy, Bitbucket Pipelines, TeamCity, C#, NodeJS, Bash, S3, ElastiCache, AuroraDB, MySQL, RabbitMQ, RavenDB, Traefik, Docker, Kubernetes, Terraform, Route53, IAM, VPC, RDS, SQS
-

Site Reliability Engineer Colorado Hosting Inc

Denver

May 2016 - April 2018

At Colorado Hosting I formulated a migration plan to move individual managed WordPress websites from running on individual virtual machines provisioned with a LAMP stack to segregated Kubernetes pods which interfaced with their own segregated databases on Google's Cloud SQL using principle of least privilege, obviously. This reduced operational overhead by approximately 30%, and while we passed this

savings onto our clients, the paradigm shift towards frameworks like React and Angular and DevOps in general slowly obliterated our clientele. Eventually I was laid off due to lack of work.

- Devised and implemented a plan for migrating about 2500 virtual machine based WordPress websites running a LAMP stack to Kubernetes and CloudSQL.
 - Crafted a pipeline to run a mysqldump job
 - Followed principle of least privilege to provide segregation of client data
 - Zero downtime live migration using a canary deployment strategy
- Reduced operational overhead by approximately 30%. \$10k/mo -> \$7k/mo
 - Achieved by reducing thousands of individual micro and small sized virtual machines to a Kubernetes cluster of about 350 medium sized nodes (2 CPU, 8G RAM).
 - Migrated thousands of mysql databases from individual VMs into a few CloudSQL databases using the principle of least privilege to segregate client data.
- Crafted and implemented SLI, SLO, and SLA policies for client sites and services.
- Integrated DataDog for monitoring. Crafted some custom alert metrics for scaling individual deployments based on latency and performance.
- **Technical Environment:** Debian, WordPress, Apache, MySQL, PHP, Jenkins, Kubernetes, GCP IAM, GCP CloudSQL, GCP Storage, GCP Networking

Embedded Software Engineer & Team Lead

Twenty Mile Security Specialists LLC

Denver & Remote

May 2010 - May 2016

Designed IOT devices in the life safety and security sector. Developed a device that could interface with major brands of burglar, fire, and medical alerting systems to allow communications over IP protocol. We were ultimately tasked with creating an environment which could relay all of the signals the devices sent from a server to client's respective monitoring agencies. Regulations in that sector prevent using a centralized server because it's a single point of failure. As the Engineering team lead, it was on me to guide the project. A friend told me about the DevOps shift and SRE. After about a year of R&D we were able to roll out an autoscaling HA Kubernetes cluster as our relay and bring the product to market.

Education

Arapahoe Community College

Littleton, CO

Associate of Science, Computer Information Systems - Programming Concentration

Graduated: May 2018

GPA: 3.97

References Upon Request