Yaqoob Ahmed

Site Reliability Engineer (SRE) · DevOps & Platform Engineer

Canada · Remote or relocation · work@ycodes.ca · ycodes.ca · github.com/yzahmed

SUMMARY

Site Reliability and Platform Engineer, freelancing since 2022 on AWS, with Cloudflare at the edge. On a 10-month contract I codified a firm's AWS environment in Terraform and Ansible and rebuilt it from code every weekday morning, cutting compute hours by ~66%; on a multi-client serverless platform, my AWS Lambda backends served 100+ clients at 99.9% uptime. Cloud-agnostic by design: the same reliability patterns on AWS, Google Cloud, Cloudflare and self-hosted Proxmox. I design for safe failure (health-check-gated automated failover, human-run failback, failover drills under load) and run my own platform lab with production discipline: Proxmox VE as code, pull-request changes, zero port-forwards and a scripted disaster-recovery drill. I also build LLM agents and MCP tool servers with least-privilege access.

EXPERIENCE

Freelance Site Reliability & Full-Stack Engineer

2022 — Present

Self-employed · Remote, Canada · Client names withheld, available on request

Platform modernization & ephemeral AWS infrastructure

Professional-services firm · 10-month contract · 2025 — 2026

- Led the firm's move off a costly proprietary software platform onto software I built in-house: a React front end and Node.js/TypeScript REST APIs that made core workflow data processing 25% faster.
- Codified the AWS environment in Terraform and configured EC2 hosts with Ansible; designed the application tier to be stateless, with all persistent data in Amazon S3.
- Engineered a scheduled infrastructure lifecycle: the environment was built automatically before the 9 a.m. business day and destroyed at 5:30 p.m. on weekdays (57.5 of 168 weekly hours), cutting compute hours by ~66% versus always-on.
- Rebuilding the environment from code every weekday proved it reproducible daily, kept configuration drift out, and left nothing running overnight.
- Stabilized production under peak load with AWS CloudWatch observability; introduced CI/CD automation that reduced manual release errors; resolved critical incidents and ran post-mortems to eliminate recurring bottlenecks.

Stack: AWS (EC2, S3, IAM, CloudWatch), Terraform, Ansible, React, Node.js, TypeScript

Cloud & SRE — multi-client serverless platform

- Architected and scaled serverless backend services on AWS Lambda and API Gateway, supporting 100+ clients at 99.9% uptime.
- Automated end-to-end deployment pipelines with Docker and Cloudflare, reducing deployment lead time by 40%.
- Engineered event-driven notification workflows that streamlined client operations.
- Secured the platform with least-privilege IAM policies and encrypted communication channels.
- Kept services highly available through proactive monitoring and rapid incident response; led the full SDLC for high-traffic dashboards.

Stack: AWS (Lambda, API Gateway, IAM), Node.js, React, PHP, MySQL, Docker, Cloudflare, Linux

SELECTED ENGINEERING WORK

- **Self-hosted platform lab, run as code** — Proxmox VE with unprivileged LXC containers and KVM guests, managed from Git through pull requests; golden images and a scripted disaster-recovery drill that restores the image from off-site storage (first run PASS in 336 s; 19-step rebuild runbook); zero port-forwards (outbound-only tunnel behind SSO, private WireGuard mesh with ACL-enforced SSH); a private Git forge; default-deny firewalls on the app stack and the forge; a health-check watchdog and state-change alerting. Case study: ycodes.ca/projects/homelab-platform
- **Self-healing reliability lab** — Kafka event bus and MySQL GTID replication behind HAProxy, with automated failover gated by three consecutive failed health checks, human-run failback, k6 load tests and written postmortems; in a failover drill under load, killing the primary cost ~10 s of write delay and no failed client requests.
  Case study: ycodes.ca/projects/sre-trading-platform

- **PKI and TLS tooling (live)** — certificate-chain inspector that verifies every signature in the chain up to Mozilla's root store; X.509 + RSA-OAEP teaching demo built from first principles.
  Case study: ycodes.ca/projects/tls-certificate-inspector
- **AI agents & MCP tooling** — LLM agents that plan multi-step tasks and act through tools; MCP (Model Context Protocol) servers giving agents scoped access to operational tooling. In my platform lab, a coding agent works on the hypervisor through a privilege-separated API token: read-only by default, full lifecycle control only inside a disposable pool, and destructive actions only with out-of-band human approval.

SKILLS grouped by capability, not vendor

Compute & serverless	AWS Lambda, EC2 · Google Cloud Run · Cloudflare Workers · Docker
Orchestration	Kubernetes (GKE, K3s), Helm · Docker Compose · Proxmox VE (LXC, KVM)
Linux & platform	Debian, systemd (services, timers), nftables, LVM-thin, rclone · Bash
Infrastructure as code	Terraform, Ansible · GitHub Actions, Gitea · Cloudflare Pages / Wrangler
Networking & edge	AWS API Gateway · Cloudflare (CDN, Tunnel, Zero Trust) · WireGuard, Tailscale · NGINX, HAProxy, Caddy
Identity & security	AWS IAM (least privilege) · OAuth 2.0, JWT · TLS, X.509 / PKI · HashiCorp Vault
Reliability & observability	SLO/SLI design, error budgets, fault-injection drills, incident response and post-mortems · Prometheus, Grafana, Loki · AWS CloudWatch
Data & messaging	MySQL (GTID replication), PostgreSQL, MongoDB, Redis · Apache Kafka, MQTT · Cloudflare KV
Languages	Go, Python, TypeScript / JavaScript (Node.js, React), Bash, SQL, PHP
AI / ML	LLM agents and tool use · MCP servers · model training and fine-tuning · least-privilege access for agents

EDUCATION

- MBA
- Postgraduate Diploma, Web Development
- Full-Stack Web Development Bootcamp