

Stephen Rhodes

Senior Software Engineer, Platform & Infrastructure

stephen@aethyriion.dev | Ohio, USA (open to relocation) | aethyriion.dev | github.com/aethyriion |
linkedin.com/in/stephen-e-rhodes

Summary

Software engineer who builds and runs the platform the code ships on. Nearly six years at an AI data platform company as a software engineer and DevOps lead: wrote Rust libraries, a Java connector and a co-designed TCP protocol for customer on-prem data, while owning the EKS clusters in CDK for Terraform, company-wide least-privilege IAM, on-call with written postmortems, and CI, where builds went from 40 to 8 minutes. On a part-time contract, designed a multi-region live-streaming platform's Rust microservices on Kubernetes (ArgoCD, PostgreSQL, Redis) and wrote the initial services.

Technical Skills

Languages: Rust, Go, TypeScript, JavaScript, Python, Java, C++, Nix

Cloud & infrastructure: AWS (EKS, EC2, IAM, S3, ECR), Kubernetes (EKS, k3s), Docker, Helm, Terraform (HCL and CDK for Terraform in TypeScript), NixOS, Nix flakes

CI/CD & observability: CircleCI (custom runners), Docker layer caching, sccache, ArgoCD, Prometheus, Grafana

Data stores: PostgreSQL, MySQL, DynamoDB, Redis

GitOps (homelab): Argo Workflows, Argo Events, Kustomize, Harbor, cert-manager, sealed-secrets

Professional Experience

Software Engineer (DevOps Lead)

Balanced Media & Technology | Austin, TX (Remote) | Nov 2020 - Aug 2026

- Software engineer and DevOps lead (senior of two, training the junior engineer into the role) on the Hewmen AI data platform: 11 services on PostgreSQL, MySQL and DynamoDB, with CI and clusters used by 12 engineers across departments.
- Connected customer on-premises data sources to the platform's broker/worker API on EKS, writing the Java connector and co-designing its TCP protocol, so OpenAI-backed LLM agents could work over private data on-prem and in uploaded documents.
- Built Aegis, the team's development environment app, in Rust (Tauri, with a Svelte UI) and Docker; its container was later reused as the CircleCI runner so local and CI builds matched.
- Wrote a headless OBS system in Go and Docker on EC2, driven by a Twitch chat bot that starts streams and switches scenes, so anyone could demo Twitch extensions live without installing OBS or handling credentials.
- Wrote Rust libraries for the main platform monorepo and contributed to several of its services.
- Operated the production and two staging Amazon EKS clusters, plus per-developer environment clusters, all defined in CDK for Terraform in TypeScript (Helm and Kubernetes providers) and deployed through CircleCI with manual approval gates.
- Cut CI builds for the main platform monorepo from 40 to 8 minutes using Docker layer caching and sccache for Rust, without carrying stale artifacts between runs; production deploys stopped taking all day.
- Owned AWS IAM for the whole company, writing every policy and role to least privilege.
- Moved Aegis to a declarative NixOS flake with pinned packages, scripts and helper tools written in Rust: onboarding on company hardware went from several days of fixing issues to nearly instant.
- Introduced Prometheus and Grafana monitoring to the platform; carried on-call for production as the lead of two and wrote every postmortem, using the same monitoring for performance analysis and AWS cost-anomaly detection.

Backend / DevOps Engineer (part-time contract)

SharePlay Interactive | Atlanta, GA (Remote) | Sep 2023 - Feb 2024

- Replaced a single Node.js application that handled one stream with a multi-user, multi-region platform of Rust microservices on Kubernetes (users, content, streams, chat, payments) on PostgreSQL and Redis, on a team of fewer than five: designed the system and wrote the initial services, including several APIs and the chat server.
- Owned the Kubernetes platform, deployed with ArgoCD, for auth, chat, stream ingress/egress and supporting services.
- Built a service-health system with active checks, Slack alerts and failure escalation, and carried on-call for it.
- Traced a roughly 2-hour outage, first flagged by monitoring, to an inherited SSL certificate renewal process and rebuilt the renewal flow from scratch.
- Built CircleCI pipelines that build, test and deploy the Vue/Nuxt front end, the Rust microservices and the Rust streaming servers, replacing manual AWS deployments.

Software Engineer

Independent | Remote | Jan 2020 - Oct 2020

- Full-time: designed and built server software, APIs and tooling in TypeScript, JavaScript and Lua for online communities running 200+ concurrent users; published fivem-webbed, an Express-style HTTP router, as open source.

Lead Software Engineer

Selfscale | Ohio (On-site) | Jan 2018 - Dec 2019

- Sole software engineer, working directly with the founder: designed and built a truck weigh-in product end to end and deployed it at a live customer site, giving operators real-time access to weigh-in data from the cloud.
- Wrote the C++ and Python firmware that reads RS232 truck-scale output on a microcontroller, the Go backend on MySQL, a Vue and TypeScript web front end and a Vue mobile app.

Selected Projects

homelab - Single-node k3s platform rebuilt from one NixOS flake; about 40 workloads delivered by an ArgoCD app-of-apps (Kustomize, Helm, sealed-secrets, a cert-manager private CA).

aethyriion.dev/projects/homelab

- Delivery pipeline: a git push runs through Argo Events and Argo Workflows (Kaniko builds) to a private Harbor registry, and the image bump lands as a revertible Git commit that ArgoCD syncs.
- Runs local LLM inference with Ollama on an RTX 3080 alongside the other workloads, so models can be tried without sending data off the machine.
- Packaged k3s 1.34.1 as a custom Nix derivation to pick up an upstream CNI startup-race fix before nixpkgs shipped it.

Education

High School Diploma