KEVIN 'TYLER' COX

Senior Backend Engineer (TypeScript/GraphQL, Document DBs)

(843) 718-4024 kevincox103@gmail.com Charleston, SC

github.com/Ty-lerCox linkedin.com/in/tyler-cox-1715065a/ ty-lercox.github.io/portfolio/

Summary

Senior backend engineer with TypeScript/Java/C# service development, GraphQL API design, and document databases (DynamoDB, Firestore). Strong serverless background (GCP Cloud Functions; familiar with AWS Lambda) with CI/CD and SRE practices (Grafana/Prometheus/Loki/Tempo). Built internal tools and reliable event-driven pipelines (CDC→Kafka); security-first with Keycloak/OIDC, PKI, and TLS/mTLS. US Eastern; 6+ hours overlap with Pacific time; comfortable with occasional travel.

Education

Computer Science

Trident Technical College

Backend & Cloud

TypeScript (backend), GraphQL (schema design & resolvers), Document DBs: DynamoDB (single-table concepts), Firestore, Serverless: GCP Cloud Functions; AWS Lambda (familiar), REST API design, Data modeling (NoSQL & relational), Event-driven pipelines (CDC → Kafka)

Security & Ops

Keycloak (OIDC/SAML), Kerberos, PKI/Certificate Authority, TLS/mTLS, Grafana, Prometheus, Loki, Tempo (SLOs, alerting, traces), Runbooks & incident response, Kubernetes (support), Ansible (provisioning)

CI/CD & Tooling

GitHub Actions, GitLab Runners, Azure DevOps, Feature flags, Testing & quality gates, Internal tools/admin consoles

Front-End (supporting)

Work Experience

Expediters International (Expediters) Aug, 2014 - Present

Software Engineer (Backend-leaning, Angular Platform)

Design and delivery of services and platform tooling across internal applications; emphasis on observability, automation, and secure-by-default practices.

- Delivered RESTful services in Java Spring Boot and C#/.NET to support logistics workflows and front-end applications.
- Modeled operational data for streaming via CDC → Kafka; established medallion layers (bronze/silver/gold) to improve lineage, SLAs, and reprocessing.
- Implemented identity and transport security with Keycloak (OIDC/SAML), Kerberos, internal CA/PKI, and TLS/mTLS.
- Built observability with Grafana/Prometheus/Loki/Tempo, SLO dashboards, and alerting; authored runbooks and supported incident response.
- Standardized CI/CD with GitHub Actions and GitLab Runners; automated environment provisioning with Ansible; supported Kubernetes deployments.
- Created internal tools/admin views for operations teams to manage workflows and data quality.
- Collaborated across product/design/engineering, providing estimates, highlighting risks/mitigations, and leading code/design reviews.

Projects

ODEYA.app Feb, 2025 - May, 2025

ty-lercox.github.io/portfolio/posts/odeya-overview/

Personal web app that auto-curates YouTube playlists from collections of channels so new uploads flow into a single playlist.

- Implemented server-side rendering for SEO; componentized TS code; reduced manual upkeep via aggregation logic.

Backend Patterns — GraphQL + Document DB

TypeScript GraphQL APIs with schema-first design and resolvers backed by document databases.

- Data modeling in DynamoDB (single-table concepts) and Firestore; serverless functions (primary: GCP Cloud Functions; familiar: AWS Lambda); added metrics hooks compatible with Prometheus/Grafana.

Observability Enablement

Service health/SLO dashboards with Prometheus metrics, Loki logs, and Tempo traces; alerting integrated with deployments.

- Trace-led debugging across services; consistent dashboards; seeded runbooks and escalation paths.

CI/CD Blueprint

Angular (Signals, RxJS, NgRx), UI architecture & component design

Languages

English

Reusable pipelines (GitHub Actions/GitLab Runners) and Ansible roles for consistent build/test/deploy across on-prem and cloud targets.

- Reduced lead time to change; standardized quality gates and environment promotion; improved reliability.

Data Flow Modernization

Database-level CDC streaming into Kafka, organized into bronze/silver/gold layers.

- Improved reliability, lineage, and downstream transformations; clearer SLAs and reprocessing.