

Karthik B

karthikbalasubramanian08@gmail.com | kaizakin.site | linkedin.com/in/balasubramanian-karthik | github.com/kaizakin

SUMMARY

B.Tech CSE student building Go (Golang) backends, container runtimes, and Kubernetes tooling. CNCF Shipwright org member and reviewer; open-source contributor focused on distributed systems, CI/CD, and cloud-native developer experience.

EDUCATION

Lovely Professional University

Bachelor of Technology in Computer Science and Engineering

Jalandhar, India

Aug 2023 – May 2027 (Expected)

EXPERIENCE

Org Member & Reviewer

Feb 2026 – Present

CNCF Shipwright (Kubernetes-native container image builds)

Remote

- Contributed to Shipwright, a CNCF project for building container images on Kubernetes using cloud-native build tools
- Authored 11 merged PRs, shipping user-facing CLI features, improving Build controller, and enhancing the developer experience
- contributions: shp buildrun gather diagnostic CLI, cross-executor build diagnostics, simplified Build configuration, Build controller reconciliation fixes, Kubernetes dynamic client support, standardized CLI output, and production-grade tests
- Tech Stack: Go, Kubernetes, controller-runtime, Cobra CLI, client-go, Tekton, GitHub Actions

PROJECTS

Event Ingestion and Notification System | Go, Kafka, Kubernetes, Helm, gRPC, PostgreSQL, SQLC, Docker, Chi

[GitHub](#)

- Architected a distributed event-driven notification system with asynchronous processing across 4 backend services
- Implemented reliable event delivery using DLQ recovery, retry mechanisms, and end-to-end event tracking
- Optimized event ingestion with concurrent worker pools, keeping request processing non-blocking under load
- Built typed inter-service communication using gRPC and Protocol Buffers for low-latency service interactions

Container Runtime Implementation in C++ | C++, Linux, Namespaces, Cgroups v2, OverlayFS, iptables

[GitHub](#)

- Built a Linux container runtime from scratch, implementing process, filesystem, and network isolation without Docker
- Implemented resource isolation, copy-on-write filesystems, and secure root filesystem switching using kernel features
- Built bridge-based container networking with virtual Ethernet pairs and NAT, enabling internet access for isolated containers
- Implemented process cloning with clone() and execve(), synchronizing parent-child execution during container startup

Real-Time Collaboration Platform | Next.js, WebSockets, Redis, PostgreSQL, Yjs, Konva.js, Rough.js, WebRTC

[GitHub](#)

- Built a production-inspired collaboration platform enabling real-time chat, collaborative editing, file sharing, and video calls
- Developed low-latency collaboration with Redis caching, chat replay, and automatic state recovery
- Engineered CRDT-based collaborative editing with Yjs for conflict-free real-time document synchronization
- Leveraged Ion SFU to power low-latency multi-user video calls through selective media forwarding

Persistent CLI Execution Store (Kin CLI) | Go, SQLite, fzf, Cobra, Bubble Tea, Huh

[GitHub](#)

- Built a terminal-first command library with SQLite and fzf for instant command search and execution
- Implemented shell-inspired placeholder parsing with interactive parameter substitution for reusable command templates
- Designed a persistent command store with deduplication, usage tracking, and efficient SQLite-based indexing
- Integrated editor-driven command authoring and fuzzy finder workflows to accelerate developer productivity

ACHIEVEMENTS

- Shipped Kin CLI via Homebrew Tap and Scoop, onboarding 50+ users within the first week
- Active technical blogger writing about low-level systems, Linux internals, and Go

SKILLS

Languages: Go, TypeScript/JavaScript, C++

Frameworks: React, Next.js, Node.js, Express, Tailwind, Cobra

Databases/ORMs: PostgreSQL, MongoDB, Prisma, SQLite

Systems & Messaging: Redis, Kafka, Docker, Kubernetes

Infra/Tools: Git, GitHub Actions, Linux, gRPC, REST, WebSockets, Helm