Vivek Kumar

Noida, India | +91-7088980706 | vivekumar2003bsr@gmail.com | linkedin.com/in/vivekumar08 | github.com/vivekumar08

Summary

Software Engineer with **3 years** building **distributed systems, event-driven pipelines, and search infrastructure** at scale using **Go, Node.js, Python, and AWS**. Full ownership from design to production — proven in fault tolerance, data consistency, and multi-tenant backend architecture.

Experience

Software Engineer Jul 2023 – Present
Bytive Technologies *Noida, India*

- Cut deployment time **40%** by automating AWS CloudFormation provisioning and CI/CD pipelines, eliminating manual overhead across microservice infrastructure (PC Jewellers).
- Built **event-driven SQS pipelines** processing **10K+ SKUs** with retry mechanisms, dead-letter queues, and eventual consistency guarantees across distributed services (Kisna Diamonds).
- Integrated **Azure Service Bus** for omni-channel order sync across ERP and e-commerce systems, enabling reliable cross-platform event streaming with guaranteed delivery (PC Jewellers).
- Reduced API latency **40%** by replacing synchronous flows with **Redis-backed async task queues**; deployed services as **AWS Lambda** functions via Serverless Framework for cost-efficient scaling.
- Engineered **hybrid vector + full-text search** in Go with OpenSearch and concurrent goroutine ingestion pipelines, improving query relevance and indexing throughput at scale (SearchByte).
- Built end-to-end **observability pipeline (Winston → Loki → Grafana)**: implemented a custom batched Loki transport with keep-alive TLS pooling, 500-entry ring buffer, and graceful-shutdown flush — used structured logs to diagnose data race conditions under concurrency without downtime.

Projects

go-s3-lite — Distributed Object Storage *Go, gRPC, Consistent Hashing*

- S3-compatible distributed storage with **consistent hashing, replication, and node-failure recovery** — durability and availability preserved under concurrent writes and node churn.
- Built concurrent metadata services and data pipelines in Go; stress-tested partition tolerance and replication lag under simulated failure scenarios.

LLM Retrieval System (RAG) *Python, FAISS, LLM Orchestration*

- Hybrid vector + BM25 retrieval with **evaluation pipelines benchmarking latency, recall accuracy, and cost** across varying corpus sizes and query loads.
- Implemented orchestration workflows simulating production traffic; validated latency-vs-relevance trade-offs by tuning chunk size and top-k retrieval parameters.

Technical Skills

- **Languages:** Go, Python, TypeScript, JavaScript, Node.js
- **Backend & Frameworks:** Node.js, Express.js, Next.js, Serverless Framework (AWS Lambda), Django, gRPC, REST APIs
- **Cloud & Infrastructure:** AWS (SQS, S3, SNS, Lambda, EC2, API Gateway, CloudFormation), Azure Service Bus, GCP, Docker, CI/CD
- **Databases & Storage:** MongoDB, PostgreSQL, MySQL, MSSQL, Redis, OpenSearch, FAISS, Sequelize
- **Frontend:** React, Next.js, Redux Toolkit, Zustand, Tailwind CSS, TypeScript
- **Observability & Reliability:** Winston, Loki, Grafana, Sentry, Logtail, Fault Tolerance, DLQs, Retry Strategies, Performance Tuning

Education

Cluster Innovation Centre, University of Delhi New Delhi, India
B.Tech in Information Technology & Mathematical Innovations ***CGPA: 8.981*** *Nov 2020 – Aug 2024*