Siva Sambath Mudaliar

siva.s.monster@gmail.com | (678) 404-9930 | Lilburn, GA, 30047 | https://github.com/siva-s-monster

Profile

Go **Developer** with attention to detail and performance focus. Hands on experience building scalable backends, **REST/gRPC** APIs, cloud native microservices. Good at writing clean Go code and deploying to **AWS** and **Kubernetes**. Proficient in database integration, CI/CD automation and troubleshooting production issues. Love delivering high performance solutions in fast paced agile teams.

Education

BACHELOR OF COMPUTER SCIENCE IN UNIVERSITY OF KENT | AUG 2009

Skills & Abilities

Golang	+10	Kubernetes	+7	AWS	+7
Python	+10	Typescript	+7	GCP	+3
React	+7	Kafka	+8	Lambda	+7
Node	+7	AI/ML	+6	SQL/NoSQL	+7
Docker	+7	CICD	+8	Java	+3

Experience

SR. GO DEVELOPER | AIRISTA (SPARKS, MD) | *JAN 2023 – PRESENT*

- Designed and developed high-performance microservices in Go, powering scalable, event-driven architectures across distributed systems.
- Migrated legacy Python services to Go, improving throughput, memory efficiency, and concurrency handling.
- Built and optimized RESTful and gRPC APIs in Go and Node.js, reducing latency, improving error handling, and ensuring secure, production-grade deployments.
- Developed a Slack GenAI bot integrated with internal APIs for summarization and Q&A over enterprise data, powered by a retrieval-augmented generation (RAG) pipeline using OpenAI and Pinecone.
- Implemented Kafka producers, consumers, and streams in Go, enabling real-time event sourcing, schema validation, and enrichment for distributed messaging.
- Built React dashboards and components to visualize service metrics, Kafka message traces, and data pipeline health in real time.
- Modernized JavaScript codebases by migrating to TypeScript, introducing strict typing, generics, and shared interfaces for improved reliability and maintainability.
- Used Databricks to train and deploy AI/ML models that powered recommendation systems and BI dashboards.
- Containerized and deployed Go and Python services on Kubernetes (EKS) with Helm, plementing observability via OpenTelemetry and Prometheus/Grafana.

- Automated CI/CD pipelines in Jenkins and GitHub Actions, covering testing, builds, security scans, and deployments with rollback and approval workflows.
- Leveraged AWS services (Lambda, API Gateway, DynamoDB, S3, IAM, CloudWatch, VPC) to build secure, serverless, and cloud-native solutions.
- Authored custom Kubernetes operators in Go, automating provisioning and cleanup for data pipelines and workloads.
- Refactored ML model wrappers from Python to Go microservices, cutting inference latency and reducing resource overhead.
- Built CLI tools in Go to replace brittle shell scripts, improving developer experience and maintainability.
- Integrated Databricks feature stores and pipelines with Go microservices for consistent ML model training and inference.
- Created reusable Terraform modules and managed multi-environment infrastructure as code, enforcing policies and drift detection.
- Migrated legacy infrastructure to AWS CloudFormation, templating VPCs, ECS/EKS clusters, IAM roles, and ALBs for multi-account rollouts.
- Designed versioned GraphQL schemas with input validation, caching, and IAM/JWT-based authorization for performant, type-safe APIs.
- Led internal workshops on Golang, Kafka, CI/CD, and cloud-native service design, fostering best practices across teams.

SR. GO/PYTHON DEVELOPER | TTEC (AUSTIN TX) | *FEB 2020 - DEC 2022*

- Built Golang backend services for real-time VoIP communications using SIP call protocols.
- Created real-time audio translation system using WebRTC, media streaming and cloud-based speech APIs.
- Implemented bi-directional streaming with gRPC in Go for low-latency audio processing pipelines.
- Designed and maintained Kafka producers and consumers for high-throughput audio metadata and transcription events.
- Added Prometheus to Go services to expose real-time call metrics like jitter, latency and packet loss.
- Created detailed dashboards in Grafana to monitor SIP call health, translation latency and system availability.
- Tuned SQL queries and indexes for Postgres-based analytics store for call duration, translations and errors.
- Built gRPC-based microservices in Golang for streaming translation workflows with resilience and failover logic.
- Implemented SIP INVITE and ACK handling in Go for initiating and terminating real-time calls.
- Used Kafka topics to stream multilingual transcription logs for post-call analytics and quality scoring.
- Monitored gRPC call performance with custom metrics and visualized real-time latency breakdowns in Grafana.
- Developed and maintained scalable backend applications in Java with clean architecture and efficient data handling.
- Implemented RESTful APIs in Java and integrated with relational databases for performance and reliability.
- Applied object-oriented programming principles and design patterns in Java for code maintainability and scalability.
- Built test frameworks to simulate SIP call traffic and test system performance under load.
- Created dynamic call routing engine in Go that adjusts based on user language, region and network conditions.
- Stored user profiles and call history in SQL database with ACID compliance and audit logging.

- Designed Prometheus alerts for gRPC failure rates and Kafka consumer lag to reduce incident response times.
- Secured React routes and API calls with JWT for data protection.
- Optimized React app performance with lazy loading, memoization and efficient rendering.
- Collaborated in Agile teams to develop and deliver scalable, production-ready React applications aligned with evolving business goals.
- Built reusable React components and custom hooks in TypeScript, improving maintainability, test coverage, and developer productivity.
- Established comprehensive tooling with ESLint, Prettier, ts-node, and CI pipelines to maintain code quality and enable safe refactoring
- Implemented secure SIP and gRPC communication channels using TLS and mTLS across audio translation services.
- Developed a Go-based voice translation middleware integrated with cloud TTS and ASR providers, enabling real-time voice translation
- Aggregated real-time SIP signaling and RTP metrics into Prometheus for accurate SLA monitoring and performance insights.
- Integrated audio playback and language detection within SIP sessions to deliver seamless multi-language user experiences.
- Conducted code reviews and performance optimizations for Golang services handling high-concurrency SIP and gRPC streams.

SR. GO DEVELOPER | UBER (SAN FRANCISCO, CA) | *JUN 2013 - JAN 2020*

- Led critical incident management efforts, coordinating triage, root cause analysis, and cross-team communication to minimize downtime and improve reliability.
- Designed and developed scalable gRPC microservices in Go to power internal APIs and high-throughput backend operations.
- Built long-running Cadence workflows for payment processing, introducing robust retry and failure-handling logic.
- Tuned PostgreSQL queries and indexes to improve transactional and reporting performance.
- Integrated Kafka producers and consumers in Go for real-time, event-driven communication across distributed systems.
- Embedded observability into gRPC services with interceptors for logging, metrics, and distributed tracing using OpenTelemetry.
- Managed and maintained services within a monorepo, enforcing shared code standards, dependency management, and consistent build pipelines.
- Authored and standardized incident response playbooks, reducing recovery time during major production outages.
- Built Go-based CLI tools for interacting with Cadence workflows, improving developer experience and debugging efficiency.
- Automated alerting and monitoring with Prometheus and Grafana, using gRPC failure rates and Kafka lag as key signals.
- Migrated legacy REST services to gRPC, improving latency, payload efficiency, and schema enforcement.
- Designed Kafka schema evolution strategies using Protobuf and Avro for safe, backward-compatible data exchange.
- Contributed to build and CI/CD improvements via monorepo linting, pre-commit hooks, and shared deployment templates.

- Led and participated in game-day simulations to validate incident readiness and refine operational procedures.
- Created database migration scripts and schema versioning workflows using Flyway and Liquibase.
- Developed gRPC middleware in Go for centralized authentication, rate limiting, and structured logging.
- Refactored cron-based jobs into Cadence workflows, improving visibility, error handling, and system reliability.
- Coordinated post-incident reviews to identify systemic issues and implement long-term solutions.
- Leveraged SQL and Kafka Streams to materialize real-time data views and support operational dashboards.

APPLICATION DEVELOPER | HUMANA (LOUISVILLE, KY) | *JAN 2010 - MAY 2013*

- Developed scalable REST APIs in Golang to handle high-throughput client requests with minimal latency.
- Built AWS Lambda functions in both Golang and Python to support serverless event-driven workflows.
- Designed NoSQL schemas and aggregation pipelines for MongoDB, optimizing for flexible and fast queries.
- Integrated Go-based microservices with MongoDB using official drivers and connection pooling for performance.
- Created data ingestion pipelines in Python to process external API data and persist it into MongoDB.
- Wrote comprehensive unit and integration tests for RESTful APIs in Golang using httptest and testify.
- Deployed stateless Golang services as containerized applications behind AWS API Gateway and Lambda.
- Developed reusable Python scripts for ETL, log parsing, and batch processing tasks in CI/CD pipelines.
- Designed and maintained REST endpoints for authentication, user management, and transaction processing.
- Built CRUD operations and middleware in Go using mux, chi, and native HTTP packages.
- Leveraged MongoDB Atlas and its metrics dashboard to monitor usage, indexing, and replica set performance.
- Used AWS Lambda with API Gateway to rapidly prototype and deploy microservices with minimal ops overhead.
- Refactored monolithic Python backend into modular Golang services to reduce deployment bottlenecks.
- Employed concurrency patterns in Go (goroutines and channels) to parallelize REST API request handling.
- Built authentication and authorization layers using JWT in Golang REST services.
- Implemented input validation and custom middleware in Python Flask apps used for internal tooling.
- Used Golang's encoding/json and net/http packages to build performant, standards-compliant APIs.
- Managed data migrations and schema evolution in MongoDB for versioned service releases.
- Created dashboards for service metrics and API usage patterns using custom logging in both Go and Python.
- Conducted peer reviews and mentored junior engineers on building clean, testable APIs in Golang and Python.