RamChand Perala | Golang Developer

+1 925-390-1137

[**chandperalaram@gmail.com**](mailto:chandperalaram@gmail.com)

# PROFESSIONAL SUMMARY:

* 6+ years of experience developing and testing **Golang-based microservices** in distributed, **cloud-native** environments.
* Designed and implemented **asynchronous network programming** patterns in **Go** for high-throughput and low-latency applications.
* Used Go frameworks like **Gin**, **Beego**, and **Mux** for building **RESTful APIs** and web services.
* Developed scalable backend services and API gateways with performance optimization using **Vert.x** and **Go**.
* Strong hands-on knowledge of **network management protocols** including **SNMP**, **gNMI**, and **gRPC**.
* Built **network tools and simulators** for real-time monitoring and control using **Go** and **Python**.
* Developed **edge network applications** with security-focused testing, simulation of attacks, and traffic validation.
* Built backend **automation tools** and scripts using **Python** for data processing, logging, and monitoring.
* Integrated **Python-based microservices** into cloud systems and optimized data pipelines using **multi-threading** and **asyncio**.
* Created automated test scripts using **Python**, **Cucumber**, **Gherkin**, and **PyTest** to streamline QA.
* Developed, containerized, and deployed applications using **Docker** and **Kubernetes**.
* Managed microservices in **AWS** (**EC2**, **Lambda**, **S3**, **RDS**, **DynamoDB**, **SQS**, **SNS**) and **Azure** environments.
* Built and maintained **CI/CD pipelines** using **Jenkins**, **GitHub Actions**, and integrated with Docker registries.
* Integrated **Prometheus**, **Grafana**, and **Loki** with backend services to enable real-time observability.
* Configured custom **metrics**, **dashboards**, and **alerts** to monitor application health and performance KPIs.
* Designed and optimized data models in **PostgreSQL**, **MySQL**, **MongoDB**, and **Couchbase**.
* Wrote complex **PL/SQL** stored procedures, triggers, and views for transaction-heavy systems.
* Implemented **Kafka-based messaging** systems using **Producer/Consumer APIs** and distributed log management.
* Designed secure, scalable **REST APIs** with **JWT-based authentication** and role-based access control.
* Performed **edge-level security testing** and penetration simulations on **Go** applications.
* Used **Swagger** and **OpenAPI** to document APIs and support service onboarding.
* Developed UI components using **Angular**, **React**, and **Bootstrap**, integrating with backend services via **REST APIs**.
* Conducted **Test Driven Development (TDD)** using **Ginkgo**, **Testify**, and **Postman** for API testing.
* Worked in **Agile Scrum** teams, participating in sprint planning, reviews, retrospectives, and daily stand-ups.
* Collaborated cross-functionally with developers, QA engineers, and DevOps to ensure high-quality and timely delivery.
* Applied **Infrastructure as Code (IaC)** concepts to automate deployments and improve environment consistency.
* Engineered **event-driven architectures** using **Go** and **Kafka**, enabling real-time data flow and efficient message processing across distributed systems.
* Designed **resilient microservices** with circuit breakers and retry logic to handle network failures gracefully.
* Developed **Go-based CLI tools** for internal debugging, monitoring, and automating routine network checks.
* Built **custom network daemons** using Go to manage telemetry data from edge devices and routers.
* Created **Python automation scripts** to manage infrastructure tasks, including provisioning resources and environment clean-up.
* Integrated **Go services** with **AWS Lambda** and **API Gateway** to build lightweight serverless APIs.
* Used **Python Fabric** and **Ansible** for automating deployment and configuration of network services.

**TECHNICAL SKILLS:**

| Category | Skills/Technologies |
| --- | --- |
| Operating Systems | Unix, Linux, Windows, Ubuntu, Mac |
| Languages | Golang, Java, SQL, C, gRPC, Protocol Buffers, Go-Swagger,Python |
| Web Technologies | HTML, AJAX, CSS3, Bootstrap, Node.JS, React JS, Angular, XML, DOM, Beego Framework, RESTful Services, gRPC, SNMP Agents, gNMI |
| Cloud | AWS (S3, Lambda, Elastic Load Balancer (ELB), IAM), Azure, Infrastructure as Code (IaC) |
| Database | Oracle, SQL Server, MongoDB, Postgres, Redis, GORM, Elasticsearch, Logstash |
| Scripting | Windows PowerShell, Shell Script, Bash, Go Libraries (go-fmt, go-vet, go-dep) |
| CI/CD | Jenkins, Git, Docker, Kubernetes, Prometheus, Loki, Grafana, Grafana-plugin-sdk-go, Automation Tools, Git, Test Automation (Ginkgo, Gomega, Testify), CI/CD Pipelines |
| Logging & Monitoring | Prometheus/client Golang, Splunk, Loki, HTTP Event Collector (HEC), Log4J, Elasticsearch, Grafana, Grafana-plugin-sdk-go, KPIs (Flight Punctuality, Baggage Handling, Customer Service Metrics) |
| Security | SAML, JWT Tokens, Security Monitoring, Compliance Monitoring, AWS Identity and Access Management (IAM) |
| Development & Testing | Docker, Jenkins, gRPC Security, Microservices Architecture, Unit Testing (Ginkgo, Gomega, Testify), Containerization, CI/CD Automation, Manual Testing, Performance Bottlenecks, Edge Testing, AWS Identity and Access Management (IAM), Distributed Systems |
| DevOps & Automation | Jenkins, Git, Docker, Kubernetes, CI/CD, Infrastructure as Code (IaC), Cross-Functional DevOps Teams, Concurrency Issues, Automation Tools, Containerization, Logging and Monitoring Infrastructure |

**WORK EXPERIENCE**

**Client Name – American Airlines, Dallas, Texas January 2024 – Current**

**Role – Golang Developer**

**Responsibilities –**

* Designed and implemented **scalable, asynchronous microservices** using **Golang**, following REST and gRPC protocols to enhance system efficiency, maintainability, and performance.
* Developed and maintained **gRPC and RESTful services** using Go frameworks like **Beego** and **Gin**, enabling robust internal communication between microservices.
* Built **SNMP agents and managers** in Go to support **real-time network monitoring**, improving observability in distributed infrastructure.
* Developed and deployed **gNMI clients/servers** using **protocol buffers and gRPC** to enable configuration and telemetry exchange between cloud platforms and edge network devices.
* Integrated Go services with **Prometheus** for custom metrics exposure and with **Grafana** for real-time monitoring dashboards, improving uptime for critical operations.
* Implemented **structured logging** and log aggregation using **Loki**, and enhanced operational intelligence by integrating logs into **Splunk** via HTTP Event Collector (HEC).
* Interfaced Go applications with **Elasticsearch** using olivere/elastic for centralized logging and implemented **Logstash pipelines** for security event tracking.
* Built backend plugins and custom data sources for **Grafana using Go SDK**, offering real-time visualizations of performance and business KPIs.
* Designed and implemented **CI/CD pipelines** using **Jenkins**, integrating Git and Go-based tooling to automate builds, tests, and production deployments.
* Utilized **Go's concurrency model** to develop high-throughput backend services with enhanced reliability and performance across airline and network applications.
* Developed CLI utilities and infrastructure automation tools using **Python**, supporting system diagnostics, build processes, and configuration management.
* Used **Docker** and **Kubernetes** for containerizing Go services, enabling scalable deployment and orchestration across multi-node clusters.
* Created **OpenAPI specifications** and generated server-side code using **go-swagger** for standardized API-first development and documentation.
* Secured microservices using **JWT** and **SAML-based authentication**, ensuring compliance with enterprise security standards.
* Managed **AWS cloud deployments**, including EC2, Lambda, ELB, S3, and IAM, to host and secure distributed systems in production.
* Wrote automated scripts and performed CRUD operations using **MongoDB**, integrated with Go via **GORM**, and supported data integrity across NoSQL/SQL databases.
* Developed **Kafka-based ingestion microservices** in Go for streaming real-time operational data, improving system responsiveness for analytics.
* Implemented **unit testing and integration testing** with **Ginkgo, Gomega, and Testify**, supporting test-driven development and CI automation.
* Applied **Go linters and vet tools** (go-fmt, go-vet, race detector) to ensure code quality, avoid concurrency bugs, and maintain consistent standards.
* Worked in **cross-functional DevOps teams**, applying **Infrastructure as Code (IaC)** principles and developing automation in Go and Python for deployment workflows.
* Contributed to **Agile ceremonies** including daily standups, sprint planning, and retrospectives, ensuring alignment and delivery in iterative development cycles.
* Executed test cases manually during UAT phases, tracked issues via **JIRA**, and coordinated with QA to maintain release quality.

**Client Name – CVS Health Care, Atlanta, GA April 2023 – December 2023**

**Role – Golang Developer**

**Responsibilities –**

* Developed **event-driven backend systems** in **Go** to handle high-volume telemetry ingestion from distributed IoT and edge devices.
* Engineered **custom TCP and UDP network protocol handlers** in Golang to enable efficient data transport and packet inspection across secure communication layers.
* Collaborated with network engineering teams to embed **Go-based agents** within infrastructure tools for **real-time network diagnostics and routing analysis**.
* Wrote **Python-based network testing utilities** to simulate protocol exchanges (SNMP/gNMI) and benchmark system throughput under load.
* Integrated **gRPC bi-directional streaming** in Go for low-latency communication between critical infrastructure components in cloud environments.
* Created **Go microservices with timeout, retry, and circuit breaker patterns** to improve fault tolerance and error handling in distributed applications.
* Used **Python scripts for log parsing and anomaly detection**, applying pattern-matching algorithms to monitor security events across network flows.
* Developed **middleware in Go for service-level authentication**, including token validation and encrypted context propagation across multiple APIs.
* Designed a **distributed tracing solution** using Go and OpenTelemetry to monitor and visualize request flows across dozens of microservices.
* Built and deployed **custom CLI tools in Golang** for DevOps automation, enabling engineers to manage infrastructure via command-line tasks securely and efficiently.
* Integrated **Amazon SQS and SNS** with Go-based services to support asynchronous job dispatch and notification handling across systems.
* Applied **static code analysis tools** in Go (such as GolangCI-Lint) to enforce secure coding guidelines and detect performance issues pre-deployment.
* Created a **multi-threaded job scheduler in Go** for running recurring background tasks, reducing latency and improving reliability of periodic workloads.
* Developed **Python-based test harnesses** to validate Go APIs against compliance scenarios, ensuring interoperability across vendor networks.
* Managed **multi-region AWS deployments** using Boto3 (Python SDK), automating resource provisioning and validating IAM policy compliance.
* Created **dynamic configuration services in Go** to centralize feature flags and environment toggles, reducing redeployment time and improving flexibility.
* Built **self-healing container services** in Go that leveraged Kubernetes liveness and readiness probes to monitor service health and trigger restarts.
* Delivered production-grade solutions using **Go's net/http and crypto/tls libraries**, securing inter-service communications with mTLS and custom cert chains.
* Participated in security audits and implemented **Go-based utilities** for scanning open ports, validating headers, and enforcing HTTPS policies across services.
* Mentored junior engineers on **asynchronous programming** in Golang, guiding them through use cases involving channels, context propagation, and worker pools.
* Led the rollout of **feature-flagged releases** for critical microservices, enabling safe canary deployments and instant rollback using CI/CD pipelines.
* Configured **custom AWS CloudWatch metrics** from Go applications to track system health and auto-scale services based on memory and throughput patterns.
* Designed **schema validation layers** in Go to enforce API input/output contracts dynamically using JSON Schema and Protocol Buffers.
* Created **automated rollback strategies** for failed Go deployments, integrating health checks and alerting with PagerDuty for faster MTTR.

**Client Name – Google, Dallas, Texas March 2022 – March 2023**

**Role – Java/Golang Developer**

**Responsibilities -**

* Developed and deployed **scalable microservices** in **Golang**, leveraging lightweight goroutines and channels to process concurrent workloads efficiently.
* Implemented **gRPC-based communication interfaces** and **REST APIs** to streamline service-to-service interactions within a distributed microservice architecture.
* Integrated **SNMP protocol handlers** and **gNMI telemetry agents** in Go to support advanced network diagnostics and real-time data streaming.
* Secured distributed applications using **OAuth 2.0 flows** and **JWT tokens**, enforcing role-based access control across internal and external APIs.
* Automated build and deployment workflows using **Jenkins**, with CI/CD pipelines configured for rapid feedback and production rollouts.
* Built **serverless backend components** using **AWS Lambda**, connecting them to event sources like S3, SNS, and DynamoDB for cloud-native processing.
* Applied **containerization practices** by packaging services with **Docker**, orchestrating workloads using Kubernetes for resilient deployments.
* Authored Python utilities for **data extraction**, **report automation**, and **test validation**, supporting DevOps and QA automation tasks.
* Designed observability pipelines using **Prometheus** and **Grafana** for custom metric exposure and real-time alerting across cloud systems.
* Integrated structured logging with **Loki** and implemented request tracing across services to improve incident response and system transparency.
* Contributed to secure networking layers by simulating attacks and conducting penetration testing for Golang-based APIs running at the network edge.
* Utilized **IAM roles and policies** in AWS to enforce fine-grained security controls over API access and infrastructure components.
* Configured **API schemas using OpenAPI/Swagger** and enabled contract testing across teams for faster integration and reduced production errors.
* Created custom resource definitions (CRDs) and Kubernetes operators using Go to extend platform capabilities and automate infrastructure provisioning.
* Developed Kafka-based Go consumers and producers to ingest real-time messages, enabling data pipelines for analytics and system event monitoring.
* Built asynchronous task processors for batch operations and background jobs using Go and message queues for non-blocking system design.
* Wrote comprehensive test suites with **Ginkgo**, **Gomega**, and **Testify**, implementing behavior-driven and unit test strategies to ensure production reliability.
* Applied **Infrastructure as Code (IaC)** techniques using Python and YAML templates to provision and update cloud resources across environments.
* Participated in Agile ceremonies including sprint reviews, grooming sessions, and retrospectives to contribute to continuous delivery cycles.
* Documented deployment workflows, architecture decisions, and troubleshooting guides for internal teams and compliance audits.

**Client Name – T Mobile , India June 2018 – December 2021**

**Role – Java Developer**

**Responsibilities -**

* + - * Involved in various phases of SDLC of the application using Agile Scrum methodology.
      * Used Hibernate to do the object relational mapping between the tables and Java objects.
      * Developed web applications using Java/J2EE technologies MVC and Spring MVC.
      * Used Spring MVC for REST APIs and Spring Boot for Microservices.
      * Deployed and tested the application with Web Servers Tomcat, WebLogic.
      * Developed the application using Servlets for the presentation layer along with JavaScript for the client-side validations.
      * Integrated the Spring framework with Hibernate ORM tool to access the Oracle database.
      * Developed Api for using AWS lambda to manage the servers and run the code in the AWS.
      * Developed monitoring and logging solutions using AWS CloudWatch and AWS CloudTrail.
      * Used Cucumber to identify, document, and report defects.
      * Managed and updated Gherkin-based test documentation.
      * Designed and developed UI components using HTML, CSS, JavaScript, and bootstrap.
      * Used Maven for build framework and Jenkins and Bamboo for continuous Integration process.
      * Implemented RESTful & SOAP Web services in Service Oriented Architecture (SOA) to retrieve data from client side.
      * Worked on implementing security for REST web services using handler interceptors.
      * Developed Spring-Jersey REST resources to serve the web application with JSON data.
      * Used Junit for test cases and Log4j to print info, warning, and error data on to the logs.
      * Used JIRA to publish daily defect reports to the management.