

Santosh Koti

San Ramon, CA | (717) 395-4709 | santosh.koti@gmail.com | [LinkedIn](#) | [GitHub](#) | pomogomo.com

PROFILE

Staff Software Engineer with 15+ years building cloud-native platform infrastructure, distributed systems, and high-throughput services at scale. Experienced in designing low-latency data planes, control-plane-driven platforms, distributed caches, and data pipelines on AWS, with deep focus on reliability, observability, performance, and production operations. Hands-on technical leader who drives architecture, mentors engineers, and raises the engineering bar through design reviews, hiring, and production-quality execution.

TECHNICAL SKILLS

Languages: Go, Rust, Java, Python, TypeScript/JavaScript

Cloud & Infrastructure: Kubernetes, kubebuilder, controller-runtime, CRDs, AWS, Terraform, Docker, Linux

Distributed Systems: Kafka, NATS, gRPC, Redis, Elasticsearch, Consul, leader election, replication, consensus

Data Systems: MySQL, PostgreSQL, Parquet/Arrow, Vector stores

Leadership: System design, architecture reviews, technical mentorship, hiring, cross-team collaboration, production on-call

EXPERIENCE

Senior Staff Software Engineer · Fanatics

June 2018 – Present

Platform & Distributed Systems

Co-authored and scaled a strongly-consistent distributed in-memory data plane in Go that resolves runtime configuration, content, feature flags, A/B experiments, and contextual targeting on the synchronous request path for 1,000+ Fanatics-managed sites. The platform sustains 100M+ requests/day with sub-10ms p95 latency. Partnered with a distinguished engineer on the original replication model and Consul-elected leader-flip protocol; led major extensions across delta publication, experimentation, replication, and request-time resolution.

- **Designed and built delta publication** on top of full-snapshot replication, shipping only changed records through the existing leader-elected version-flip path. Increased publication frequency from a handful per day to hundreds per day with no latency regression — the largest throughput improvement to the platform after the initial build.
- **Designed and built a namespace-based experimentation and rollout platform** controlling user allocation, staged launches, and PlanOut-style experiment isolation across 1,000+ sites. Supports approximately 100 concurrent experiments and became the default launch path for nearly every product release.
- **Owned the in-memory configuration model and per-request resolution engine**, including nested envelopes, ranking, scheduling, overrides, labels, and contextual matching across device, geo, currency, league/team, locale, and user segments.
- **Extended the replication and snapshot path** to support delta semantics, multi-publication isolation, and expanded request-time resolution surfaces while preserving availability, consistency, and low-latency guarantees.

Reliability, Performance & Data Pipelines

- Optimized large-scale database queries, including GraphQL-generated access patterns, transaction blocks, and connection handling; resolved production deadlocks and segregated read/write traffic to appropriate replicas, improving API latency and stability.
- Reduced Elasticsearch index size by approximately 20% and substantially reduced re-indexing times through mapping and indexing optimizations.
- Owned production on-call for content publication and experimentation platforms; triaged incidents, drove postmortems, and improved observability, alerting, and incident response in partnership with SRE teams.
- Built a hybrid Kafka + Parquet ingestion pipeline (live streaming + full-refresh batch) used as a shared data substrate across teams; built a separate Parquet/Arrow reporting pipeline for efficient analytics over large datasets.
- Designed an Algolia indexing pipeline for Japanese-locale sites, supporting Kanji, Hiragana, and Katakana search behavior.

- Migrated application infrastructure provisioning to Terraform and executed major-version database upgrades via infrastructure-as-code with minimal downtime.

Technical Leadership

- Mentored two engineers to Senior and Staff levels; drove cross-team architecture reviews, participated in technical hiring (screens, on-sites, debriefs), and helped raise engineering quality across teams.

Stack: Go, TypeScript, AWS, Kubernetes, Kafka, Consul, Elasticsearch, Redis, Terraform, MySQL, PostgreSQL, gRPC, Parquet/Arrow.

OPEN SOURCE

Distributed Cache Operator — Kubernetes Operator

Go, kubebuilder, controller-runtime, Calico

github.com/GolfRider/distributed-cache-operator

- Built a Kubernetes operator for sharded in-memory cache clusters using client-side consistent hashing, CRDs, and controller-runtime reconciliation.
- Designed a control-plane-driven architecture where cache pods remain peer-unaware and ring membership is published through versioned ConfigMaps.
- Implemented drain-finalizer semantics to remove pods from the hash ring before termination, allowing clients to converge before pod shutdown.
- Designed a cell-based fault-isolation model by provisioning each cache cluster as an isolated tenant cell with dedicated namespace, ResourceQuota, and NetworkPolicy boundaries enforced through the Kubernetes API server and CNI.

EXPERIENCE (CONTINUED)

Senior Software Engineer · Visa

Jan 2018 – June 2018

- Built monthly and yearly settlement report generation over HDFS in Go; migrated calendaring logic from shell scripts to Go components; authored Cucumber/Scala test coverage for settlement workflows.

Stack: Go, Scala, Kafka, HDFS, Nomad, Ansible.

Senior Software Engineer · Continental

May 2017 – Jan 2018

- Built REST and gRPC APIs in Go for car-key provisioning and lifecycle management; authored the public API spec and error model; built a Go load-testing tool that surfaced API bottlenecks.

Stack: Go, gRPC, Protocol Buffers, PostgreSQL, Docker, Kubernetes, AWS.

Senior Software Consultant · Google

June 2016 – May 2017

- Implemented Corp SSO for PrintCentral against internal Stubby APIs; built ETL pipelines for asset management and integrations with Google Unified Ticketing System.

Stack: Python, Django, Memcached, App Engine.

Senior Software Consultant · Equinix

June 2014 – May 2016

- Migrated ECX (Equinix Cloud Exchange) to a microservices architecture; built Java/Spring Boot and Node.js services and SDN workflows on OpenDaylight APIs.

Stack: Java, Spring Boot, Node.js, REST, SQL.

Earlier Roles

- **Senior Technical Consultant · Intel** (2013–2014) — IoT sensor pipelines, notification services, dashboards. Java, Spring, Apache Camel, MongoDB.
- **Senior Software Engineer · MusicPolo** (2010–2013) — Scalable music portal: REST APIs and custom load balancer. Scala, Play, Akka.
- **Member of Technical Staff · VMware** (2008–2010) — SaaS platform services for ITIL/Compliance; vSphere API integration. Java/J2EE, Python, C++.
- **Technical Specialist · Infosys** (2003–2008) — J2EE framework benchmarking; in-house profiler development.

EDUCATION

Bachelor of Engineering, Computer Science

Visvesvaraya Technological University (VTU), India