# Ritik Chawla

ritikchawla.vercel.app — linkedin.com/in/ritikchawla — chawlaritik@gmail.com — +91 8743844765 — New Delhi, India

## Experience

**Gojek** | *Software Engineer 2* **Aug 2024 – Present**

- Developed scalable microservices in Golang handling 50K+ transactions per minute with sub-second latency.
- Architected event-driven systems using Kafka, ensuring 99.99% reliability and processing 5TB+ daily data.
- Designed robust backend APIs and optimized data pipelines handling 50M+ daily events.
- Redesigned monolith into modular microservices, improving maintainability and 2x deployment speed.
- Built versioned, backward-compatible APIs used by 10+ consumer services across teams.
- Implemented rate limiting, circuit breaking, and resiliency patterns to improve system performance under load.
- Designed efficient data models and storage strategies for high-volume transactional systems, cutting costs by 20%.
- Implemented algorithmic and database optimizations, reducing compute usage, improving response times by 35%, and cutting database latency by 40%.
- Enhanced CI/CD pipelines by adding automated integration tests, improving deployment success rates by 25%.
- Led migration from GCP to Tencent Cloud, refactoring services for compatibility and optimizing costs.
- Mentored junior engineers on best practices in API design, clean architecture, and Golang concurrency patterns.

**MAQ Software** | *Software Engineer* **July 2022 – Aug 2024**

- Built data ingestion systems consuming REST and GraphQL APIs into SQL Server and Datalake using Databricks.
- Implemented distributed caching layer, reducing average API response time by 70%.
- Optimized data warehouse queries, cutting processing time by 80% for critical business reports.
- Designed ETL pipelines processing 1TB+ daily, improving data quality and reducing downstream errors.

## Education

**Guru Gobind Singh Indraprastha University** **August 2018 – July 2022**

*Bachelor of Technology in Computer Science* ***CGPA: 9.16/10.0***

## Skills

**Languages**: Golang, C++, Python, Ruby, SQL
**Systems**: Distributed Systems, Microservices, Cloud Native Architecture, Service Mesh, System Design
**Infrastructure**: Kubernetes, Docker, Terraform, Chef, ArgoCD, AWS, GCP, Tencent Cloud
**Data and Tools**: Kafka, RabbitMQ, Redis, Vault, Elasticsearch, Prometheus, gRPC, PostgreSQL

## Projects

**Distributed Key-Value Store** | *Golang, gRPC*

- Built a distributed key-value store achieving linearizable consistency using multi-Raft consensus.
- Implemented automated leader election, log compaction, and zero-downtime membership changes.
- Achieved 50K QPS with <5ms p99 latency using custom WAL and LSM tree storage engine.
- Integrated conflict resolution with vector clocks and Merkle trees for eventual consistency.

**High-Performance Load Balancer** |*Golang, Docker*

- Implemented L7 load balancer using consistent hashing, connection pooling, and custom TCP congestion control.
- Built distributed health checking with phi-accrual failure detection and adaptive load shedding.
- Handled 100K RPS with <1ms added latency using lock-free queue design.

**Real-time Chat Application** | *Golang, WebSocket, Redis*

- Developed scalable chat service supporting 10K+ concurrent users with real-time message delivery.
- Implemented message ordering and delivery guarantees using custom sequence numbering and acknowledgments.
- Built presence system and typing indicators using Redis pub/sub with optimized memory usage.
- Added end-to-end encryption using Signal protocol for secure communication between users.

**Distributed Rate Limiter** | *Golang, Redis*

- Implemented a distributed rate limiting service using a sliding window algorithm and Redis as a backing store.
- Built coordination mechanism using Redis Lua scripts ensuring atomic operations across multiple nodes.
- Implemented token bucket and leaky bucket algorithms with configurable burst handling.