KOUSHIK GUDIPELLY

Backend / Python Software Engineer

CA, USA | gudipellykoushik@gmail.com | 619-878-7924 | LinkedIn | GitHub

SUMMARY

Backend-focused Software Engineer with 4+ years of experience designing and shipping high-throughput microservices, event-driven data pipelines, and distributed systems at Bank of America and Deloitte. Deep expertise in Python, FastAPI, Apache Kafka, and Spark, building APIs and services that process millions of transactions daily with sub-second latency. Proven track record cutting query times, shrinking deployment cycles, and shipping ML-backed backend services. Comfortable operating end-to-end across backend, full-stack, and ML/AI engineering roles.

TECHNICAL SKILLS

Languages: Python, Java, JavaScript, TypeScript, SQL

Backend & APIs: FastAPI, Flask, Spring Boot, REST APIs, GraphQL, WebSockets, Microservices, Async Programming

Data & Streaming: Apache Kafka, Spark Streaming, PySpark, ETL Pipelines, Event-Driven Architecture

Cloud & DevOps: Azure (AKS, App Services), AWS (EC2, S3, RDS, Glue, Cognito), Docker, Kubernetes, Jenkins, GitLab CI/CD

Databases: PostgreSQL, MySQL, Redis, Snowflake

ML & Analytics: XGBoost, Scikit-learn, TensorFlow, Pandas, NumPy, Feature Engineering, Anomaly Detection, CNNs

Monitoring & Security: Prometheus, Grafana, JWT, OAuth2, RBAC, SonarQube

Testing: PyTest, Mockito, Unit & Integration Testing

EXPERIENCE

Software Engineer — Backend / Python — ***Bank of America — NC, USA*** *Aug 2024 – Present*

- Architected a real-time fraud detection backend processing 1–2M daily transactions using FastAPI, Python, Apache Kafka, and Spark Streaming, achieving sub-second fraud scoring latency.
- Built and deployed ML-backed fraud scoring services (XGBoost, Scikit-learn) as FastAPI microservices, improving fraud detection accuracy by 12–15% and reducing false positives.
- Designed scalable Kafka and Spark data pipelines for real-time ingestion, transformation, and validation of high-volume financial transaction streams.
- Developed rule-based fraud detection logic (velocity checks, geo-location anomalies, device fingerprinting, behavioral profiling), increasing early detection effectiveness by 15%.
- Optimized MySQL schema design, indexing strategy, and query execution, reducing query times by 20% across core transaction tables.
- Automated CI/CD pipelines with Jenkins, Docker, and Kubernetes (SonarQube + Nexus), cutting deployment cycle time by 30% and improving release consistency.
- Implemented JWT-based authentication and RBAC across microservices, ensuring secure access control and financial compliance standards.

Full Stack Software Developer — ***Deloitte — Hyderabad, India*** *Jan 2021 – Jul 2023*

- Designed and delivered a Python/FastAPI microservices backend for an e-commerce order management platform handling 40K–50K daily transactions, improving system stability under peak load.
- Led migration of a legacy monolithic system to domain-driven microservices (FastAPI async + Spring Boot), reducing release cycles by 30% and enabling independent service deployments.
- Built RESTful and GraphQL APIs with FastAPI, Pydantic, and OpenAPI, improving response latency from ~300ms to ~200ms with strict schema validation.
- Implemented event-driven order tracking using Apache Kafka and WebSockets, reducing processing lag during peak traffic by 25%.
- Optimized backend performance via Redis caching and async processing, cutting API response latency from ~500ms to ~300ms.
- Established CI/CD pipelines with GitLab CI, Docker, and PyTest, improving build reliability, code quality, and deployment consistency.
- Deployed and monitored cloud-native backend services on Azure AKS with Prometheus and Grafana observability, maintaining high availability in production.

PROJECTS

Distributed Rate Limiter | Python, FastAPI, Redis, Docker, Prometheus

- Built a distributed API gateway enforcing token-bucket, sliding-window, and fixed-window rate limits atomically across 3 replicas via Redis Lua scripting, with zero over-limit leakage across 20k+ requests at ~400 req/s (k6-verified).
- Designed a two-tier caching strategy (in-process policy cache + shared Redis state) with a measured p50 of 0.3ms / p99 of ~9ms limiter overhead per request (44k decisions, Prometheus histogram, 3 replicas under load).
- Implemented configurable fail-open/fail-closed degradation on Redis outage; shipped with CI (GitHub Actions: ruff + pytest against a live Redis service) and full observability via Prometheus + Grafana dashboards.
- Reproducible one-command deployment via Docker Compose (8 services: 3 gateway replicas, nginx, Redis, 2 demo upstreams, Prometheus, Grafana) — verified end-to-end.

AI Resume Analyzer | React/TypeScript, FastAPI, Groq, Chroma

- Architected a full-stack AI resume screening platform (React 19/TypeScript, FastAPI, SQLite) with 7 REST endpoints, PDF/DOCX/TXT parsing, and an async scoring workflow with real-time status polling on the frontend.
- Built a RAG-based scoring pipeline combining Chroma vector search (BAAI/bge-small-en-v1.5 embeddings) with Groq LLM inference, cross-validating AI-generated skill matches against a 60+ skill taxonomy to guardrail hallucination, and chunking resumes (400-word windows, 80-word overlap) to cut LLM context per request.
- Engineered fault-tolerant LLM orchestration with automatic fallback model routing and retry-with-backoff, increasing effective request capacity 14x (1,000 → 14,400 requests/day) under rate limits.
- Optimized the embedding backend (ONNX/fastembed vs. sentence-transformers) to run within a 512MB memory ceiling, enabling low-cost cloud deployment.

EDUCATION

Master of Science in Computer Science *Aug 2023 – May 2025*

NC State University, Raleigh, NC, USA