

Gaurav Kumar

Big Data & GenAI Engineer | Pharma & Life Sciences

+91 79031 73922 | gauravsingh4frnds@gmail.com | linkedin.com/in/gaurav-kumar
github.com/gauravkumar22927 | gauravbuilds.online

Professional Summary

Big Data & GenAI Engineer with 3+ years building large-scale, fault-tolerant data platforms across **pharma & life sciences** and enterprise domains. Core expertise in **Apache Spark (PySpark)**, **Delta Lake**, and **lakehouse architecture (medallion pattern)**, with **data governance via Unity Catalog**, distributed Spark-on-Kubernetes processing, and performance tuning over multi-billion-row workloads. At ZS, engineers payer-performance, claims, and access pipelines across **Pharmacy & Medical Benefit** domains, plus data-quality gating and patient-segmentation data products. Also delivers production **GenAI** workflows – agent orchestration, RAG, and LLM integration (OpenAI, Anthropic). Strong CS fundamentals, backend/REST development, and code-review-driven SDLC.

Core Technical Skills

Languages & Querying	Python, Go, SQL, PySpark, Spark SQL
Big Data & Processing	Apache Spark (DataFrames, Structured Streaming), Kafka, Airflow, Hadoop, Hive Metastore
Lakehouse & Storage	Delta Lake (ACID, time travel, CDF, Z-ordering, liquid clustering, schema evolution), Medallion Architecture (Bronze/Silver/Gold), Unity Catalog
Cloud & Platforms	Databricks, Microsoft Fabric (OneLake, Lakehouse, Semantic Link), AWS (S3, SQS, SageMaker, Redshift), Docker, Kubernetes, Jenkins, CI/CD
Governance & DQ	Unity Catalog (RBAC, access control, column masking), custom PySpark data-quality & validation frameworks
GenAI	LLM Integration (OpenAI, Anthropic), Agentic AI, RAG, Prompt Engineering, Vector Embeddings, LangChain
Backend	REST APIs, JDBC, Microservices, Event-Driven & Distributed Systems
Data & Tools	PostgreSQL, Snowflake, Redshift, Redis, MongoDB, MySQL, Power BI, Tableau, Prometheus, Grafana, Git, Linux
Pharma Domain	IQVIA LAAD (Rx claims), MMIT payer access, Xponent/provider performance, payer lives, market-access & launch analytics

Professional Experience

Business Technology Solutions Engineer	Jan 2025 – Present
ZS Associates Pharma & Life Sciences	Gurugram, India

Payer Performance (PB & MB) – Distributed Claims & Access Pipelines

- Engineered **Pharmacy Benefit (PB)** and **Medical Benefit (MB)** payer-performance datasets over **3B+ claims records** and ~381M access records, powering both executive dashboards and self-service reporting for a pharma brand portfolio.
- Built a custom **PySpark controller-executor framework on Kubernetes**, orchestrated by **Airflow**, processing multi-billion-row claims / access / claims×access joins and landing curated tables in **Amazon Redshift** for Tableau consumption.
- Developed complex SQL data models and created a **BusinessObjects Universe** semantic layer across multiple datasets to enable governed self-service reporting.
- Authored **OPS SOP documentation** defining weekly/monthly refresh execution order and owned **dev→prod migration** for production deployment.

Launch & Market-Access Analytics – Backend Data Engineering & Data Quality Gate

- Built **backend data pipelines** in **Microsoft Fabric (PySpark, Spark SQL, Delta, OneLake)** curating payer lives, paid claims, provider performance, and MMIT access data from IQVIA LAAD, MMIT, and Xponent sources into governed Delta tables that power a **Power BI** semantic model and market-access launch dashboards.
- Designed and built a **production data-quality gate** running **~30 validations across 5 domains** (payer lives, paid claims, provider performance, MMIT access, access×performance) that **blocks the semantic-model refresh on failure**, preventing bad data from silently reaching the dashboard.
- Implemented a gating harness that tags results per suite, treats exceptions as **hard failures (no silent passes)**, writes a **run_id**-keyed audit table, and drives a JSON status payload consumed by conditional pipeline + email steps.
- Cut QC runtime **~25 min → a few minutes** by pruning raw IQVIA LAAD Parquet reads to the latest data-week partition window instead of re-deriving the pipeline from raw.
- Eliminated false-failure noise across out-of-sync pipelines by computing the reference week as the **latest week common to source and output tables** (per-domain set intersection), surfacing pipeline drift in the run log.

Power BI Usage Analytics – Backend Data Layer

- Built the backend data layer on **Microsoft Fabric (F64)** using a scheduled **PySpark + Semantic Link** notebook to extract usage telemetry into **OneLake Delta tables** for downstream dashboards.
- Engineered an **incremental Delta append/merge** pattern accumulating **6+ months of history** (vs. the native report's ~30-day retention) while refreshing only the latest ~30-day window per run – and removed the Power BI Pro license dependency for client viewers.
- Led a **Tableau → Power BI migration on Delta Lake**, applying star-schema design and **Z-order optimization** to cut query latency with full parity across 5+ dashboards.

SLE Patient Journey Analysis – Governed Lakehouse Data Product on Databricks

- Designed and built the SLE patient-journey analysis as a **governed Delta Lake data product on Databricks**, ingesting Symphony Health medical & pharmacy claims through a **medallion (Bronze/Silver/Gold)** architecture, locking an **~18K-patient SLE cohort** via a four-stage inclusion funnel, and serving Gold tables for direct consumption by field-reporting and brand-analytics teams.
- Modeled a **master patient-characteristics Silver table as the pipeline's hub**, fanning out to comorbidity (**Charlson CCI**), disease-severity, **HCRU**, and treatment-sequencing modules – a single governed source of cohort truth with no repeated re-derivation across modules.
- Derived **patient-journey archetypes** by embedding longitudinal clinical event sequences with **FastText** and clustering the latent vectors with **KMeans**, tracked and versioned in **MLflow**, writing cluster labels back to Delta so patient-level data never leaves the governed environment between SQL and Python stages.
- Engineered the pipeline as **Git-integrated Databricks Workflows** with dependency-ordered tasks and idempotent **MERGE-based loads** – mid-pipeline restart, automated lineage, and reproducible deployment built in.
- Enforced **Unity Catalog** governance across all patient-level data (RBAC, **column masking** on identifiers, row-level filters, data classification, audit logging), keeping the cohort compliant and audit-ready, and optimized Gold serving tables with **Delta partitioning and Z-ordering** on high-cardinality patient/journey keys plus pre-aggregated rollups.

Software Engineer

FourKites

Jun 2024 – Dec 2024

Chennai, India

- Architected an event-driven vessel-metadata ingestion platform with confidence-scored deduplication, publishing change events to **Amazon SQS** consumed by async workers to upsert PostgreSQL over JDBC, with DML audit logs archived to S3.
- Quantified carrier ETA accuracy at lane level by analyzing **200K+ daily vessel pings**, converting findings into shipper-facing carrier recommendations that improved provider selection and reduced logistics cost.

Data Engineer

Tata Consultancy Services (ASML Netherlands)

Jan 2021 – Jul 2022

Bangalore, India

- Built an end-to-end ML pipeline on ASML semiconductor manufacturing data – unified fragmented multi-file sensor logs into structured datasets, engineered 40+ features, and automated KPI identification for production monitoring.
- Improved XGBoost regression accuracy by 15% via recursive feature engineering, Bayesian hyperparameter tuning, and NSGA-II multi-objective feature selection.

Selected Projects

- **Open Lakehouse Platform** (*Spark 4.0, Delta Lake, Hive Metastore, Hadoop, Kubernetes, MinIO/S3*) – Production-style open lakehouse on a Kubernetes cluster, self-managing the stack Databricks and Fabric abstract away; Spark 4.0.1 SQL engine over a Hive Metastore catalog and MinIO S3 with Delta ACID tables, fully containerized with idempotent deploy scripts and end-to-end smoke tests.
- **AtlasCare – Agentic GenAI for Customer Support** (*LLM Orchestration, RAG, Tool-Calling, Async*) – GenAI agent automating Tier-1 support by orchestrating 4 enterprise backends via LLM tool-calling with a KB-driven RAG policy engine; async payment pipeline with semaphore-gated concurrency, auto-retry, event-sourced audit trail, load-tested at 50 RPS with zero drops.
- **FinFlow – Distributed Payment Processing** (*Go, Kafka, Kubernetes, Redis, Prometheus/Grafana*) – Event-driven Go microservices over Kafka with a double-entry ledger using **HOLD-based reservation** and idempotent processing to prevent duplicates under failure.

Education & Certifications

M.Tech, Computer Science & Engineering – MNNIT Allahabad (CPI 8.0/10)

2022 – 2024

B.Tech, Computer Science & Engineering – Pondicherry Engineering College (CGPA 8.07/10)

2016 – 2020

Certifications: Microsoft Azure Data Scientist Associate • Azure Data Fundamentals • Azure AI Fundamentals • Azure Fundamentals • Python Institute PCAP