ASIL KAMEPALLI

SENIOR DATA ENGINEER

San Diego, CA | +1 (331) 226-7117 | asilkamepalli9@gmail.com | linkedin.com/in/asil-kamepalli | github.com/Asil143

Python | SQL | PySpark | Apache Spark | Databricks | Snowflake | Airflow | dbt | Kafka | AWS | Azure | Delta Lake | Apache Iceberg

PROFESSIONAL SUMMARY

Senior Data Engineer with over 5 years of experience designing, building, optimizing, and operating production-grade data platforms for analytics, operational reporting, real-time decisioning, and AI/ML enablement. Possesses strong hands-on expertise in Python, SQL, PySpark, Apache Spark, Databricks, Snowflake, Airflow, dbt, Kafka, AWS, and Azure. Proven track record in batch and streaming ingestion, CDC, dimensional modeling, lakehouse architecture, data quality, governance, observability, CI/CD, Infrastructure as Code, and performance/cost optimization. Demonstrated ability to own ambiguous data problems end-to-end—from source-system analysis and resilient pipeline design to production support, root-cause remediation, and trusted data-product delivery. Possesses additional experience building governed data foundations for embeddings, vector search, and RAG workloads.

TECHNICAL SKILLS

Programming & Processing: Python, SQL, PySpark, Spark SQL, Apache Spark, Spark Structured Streaming, Bash

Cloud & Storage: AWS - S3, Glue, Lambda, EMR, Redshift, Kinesis, CloudWatch, IAM/KMS; Azure - ADF, ADLS Gen2, Databricks, Synapse, Event Hubs, Key Vault, Azure Monitor

Warehouse & Lakehouse: Snowflake, Amazon Redshift, Azure Synapse, Databricks Lakehouse, Delta Lake, Apache Iceberg, Parquet, Medallion Architecture

Orchestration & Transformation: Apache Airflow, dbt, Databricks Workflows, Azure Data Factory, AWS Glue Workflows

Data Engineering & Modeling: ETL/ELT, CDC, Incremental Processing, REST/API Ingestion, Dimensional Modeling, Star Schema, SCD Type 1/2, Data Marts

Quality, Governance & DataOps: dbt Tests, Great Expectations, Data Contracts, Lineage, Unity Catalog, RBAC/IAM, Git, GitHub Actions, Azure DevOps, Terraform, Docker, CI/CD, Observability

AI Data Infrastructure: Unstructured Data Ingestion, Embeddings, Vector Search, RAG Data Pipelines, MLflow, Feature/Training Data Pipelines

PROFESSIONAL EXPERIENCE

Publicis Sapient | Remote, USA — October 2024 - Present

Senior Data Engineer - Data Platform & AI Enablement | *Client: Comcast Advertising*

Project: Audience, Campaign & Real-Time Media Intelligence Platform - modernized campaign, audience, CRM, clickstream, impression, conversion, and reference-data processing for near-real-time analytics, attribution, segmentation, optimization, and AI-enabled use cases.

- Designed an Azure-first lakehouse platform processing 2+ TB of daily batch and streaming data, integrating relational, REST API, file, CRM, campaign, and digital-event sources into governed Bronze, Silver, and Gold data products for analytics and downstream AI/ML consumption.
- Built parameterized Azure Data Factory and Python ingestion frameworks with schema capture, audit metadata, replayability, source-specific error handling, and idempotent landing patterns in ADLS Gen2.
- Developed reusable PySpark pipelines in Azure Databricks using Delta Lake for deduplication, late-arriving data, schema evolution, SCD Type 2 history, incremental MERGE processing, and standardized business-rule execution.
- Engineered near-real-time Kafka/Event Hubs pipelines with Spark Structured Streaming handling 50M+ daily events using checkpointing, watermarking, event-time processing, dead-letter handling, replay/backfill controls, and exactly-once-oriented idempotent writes.
- Modeled curated campaign, audience, attribution, and performance marts in Snowflake with dbt, implementing incremental models, snapshots, tests, documentation, clustering strategies, and workload-specific warehouse sizing.
- Optimized Spark workloads with AQE, partition pruning, broadcast joins, skew mitigation, compaction, caching, executor tuning, and cluster right-sizing, achieving a 30% runtime reduction and 25% compute savings.
- Implemented production data-quality gates using dbt tests, PySpark assertions, source-to-target reconciliation, freshness SLAs, schema contracts, and anomaly checks so invalid data was blocked before publication.
- Established platform observability using Azure Monitor, structured logs, pipeline metrics, freshness/SLA dashboards, automated alerts, and runbooks for retries, backfills, late data, dependency failures, and incident triage.
- Built governed unstructured-data pipelines for enterprise knowledge use cases, including extraction, normalization, chunking, metadata/version enrichment, embedding generation, vector indexing, and security-filtered retrieval datasets for RAG applications.
- Partnered with analytics, data science, product, and business teams to define data contracts, resolve ambiguous source logic, review architecture, troubleshoot production issues, and mentor engineers on Spark, SQL, testing, and DataOps practices.

Environment: Azure Data Factory, ADLS Gen2, Azure Databricks, PySpark, Spark Structured Streaming, Delta Lake, Kafka/Event Hubs, Snowflake, dbt, Azure Monitor, MLflow, Terraform, Git, CI/CD.

Accenture | Remote, USA — January 2023 - September 2024

Cloud Data Engineer | *Client: Elevance Health*

Project: Enterprise Claims, Member & Provider Data Modernization - migrated legacy payer-data workloads to a governed AWS/Snowflake platform supporting claims, eligibility, member, provider, pharmacy, finance, and operational reporting.

- Modernized legacy healthcare integration workloads into an AWS-based data platform using S3, Glue, PySpark, Airflow, Snowflake, and dbt, separating raw, standardized, and consumption-ready layers.
- Built reusable ingestion frameworks for database extracts, REST APIs, partner files, and CDC feeds with partitioning, encryption, schema capture, audit columns, retries, quarantine handling, and recoverable incremental loads.
- Developed PySpark and AWS Glue transformations to cleanse, standardize, reconcile, and join high-volume claims, eligibility, member, provider, and pharmacy datasets (50M+ records) while handling duplicates, source corrections, and late-arriving records.
- Created Apache Airflow DAGs with sensors, dependencies, retries, SLAs, backfills, alerting, and environment-aware configuration to orchestrate ingestion, transformation, validation, and warehouse publication.
- Designed Snowflake dimensional models and dbt layers for trusted reporting marts, including fact/dimension structures, conformed dimensions, SCD Type 2 processing, incremental strategies, snapshots, automated tests, and documentation.
- Implemented source-to-target reconciliation for row counts, claim/payment totals, referential integrity, duplicate detection, freshness, and source-target variances before datasets were certified for analytics consumption.
- Improved Spark and Snowflake performance across multi-terabyte tables using query profiling, pruning, file sizing, clustering, materialization choices, warehouse sizing, and incremental processing, achieving a 35% latency reduction and 30% compute cost savings.
- Automated pipeline and infrastructure releases through Git, CI/CD, Terraform, code review, and controlled promotion across development, test, and production environments.
- Owned production support for failed loads and unexplained data discrepancies, tracing issues across source extracts, orchestration, transformations, and warehouse models, and converting root causes into preventive controls.
- Applied least-privilege IAM, KMS encryption, secrets management, audit logging, and traceable lineage to sensitive healthcare datasets while maintaining reusable, governed data products for downstream analytics teams.

Environment: AWS S3, AWS Glue, Lambda, PySpark, Apache Airflow, Snowflake, dbt, Terraform, CloudWatch, IAM/KMS, Git, CI/CD.

Persistent Systems | Pune, India — June 2021 - December 2022

Data Engineer | *Client: Medtronic*

Project: Global Sales, Inventory & Supply Chain Analytics Platform - integrated ERP, product, inventory, sales, distribution, and operational data to support trusted KPI reporting, fulfillment visibility, and analytics.

- Built Python and SQL ETL pipelines to integrate ERP, product, inventory, sales-order, distribution, REST API, CSV, and JSON sources (8M+ records processed daily) into analytics-ready cloud storage and warehouse tables.
- Created reusable ingestion modules for connectivity, file handling, incremental watermarks, audit logging, exception handling, and restartability, reducing duplicated logic across recurring source feeds.
- Developed SQL and PySpark transformations using CTEs, window functions, aggregations, joins, deduplication, standardization, derived attributes, and business-rule validation for reporting and downstream analytics.
- Designed star-schema marts with fact and dimension tables, surrogate keys, date dimensions, and historical attribute management to support consistent KPI definitions across sales, inventory, and operations reporting.
- Implemented source-to-target reconciliation, null/duplicate checks, schema validation, exception reporting, and controlled reprocessing to prevent bad data from reaching scheduled reports.
- Optimized Parquet layouts and relational queries through partitioning, file sizing, indexing, predicate filtering, query rewrites, and staging-table design, reducing query latency by 45% and lowering cloud storage overhead.
- Automated recurring batch jobs with dependency checks, retries, file-arrival validation, notifications, and restart procedures, improving reliability over manually triggered and one-off scripts.
- Collaborated with business users, analysts, QA, application teams, and database engineers to investigate KPI mismatches, resolve source-data issues, support releases, and document production handoffs.

Environment: Python, SQL, PySpark, PostgreSQL/SQL Server, REST APIs, Parquet, cloud object storage, Git, CI/CD, dimensional modeling, batch orchestration.

SELECTED ARCHITECTURE & ENGINEERING HIGHLIGHTS

Real-Time Event-to-Warehouse Pipeline

- Designed a reference streaming architecture using Kafka/Event Hubs/Kinesis, Spark Structured Streaming, Delta Lake/Iceberg, and cloud object storage with checkpointing, watermarking, schema evolution, deduplication, and replay support.
- Implemented Gold-layer aggregations and warehouse serving patterns for near-real-time KPIs while separating raw event retention from curated analytical consumption.

Trusted Healthcare & Enterprise Data Product Layer

- Designed a governed pattern for S3 raw landing -> PySpark/Glue standardization -> Airflow orchestration -> Snowflake/dbt dimensional marts, with source-target reconciliation, SCD Type 2 history, freshness checks, referential integrity, and lineage.

- Emphasized correctness and restartability: failed validations block publication, reconciliation differences are investigated to root cause, and backfills are idempotent.

Enterprise RAG Data Foundation

- Built a reusable data-engineering pattern for enterprise RAG workloads: ingest documents and metadata, normalize and chunk content, create embeddings, index vectors, enforce metadata security filters, and expose governed retrieval-ready datasets.
- Applied lineage, access control, validation, observability, and lifecycle tracking so AI applications consume traceable and governed enterprise knowledge rather than unmanaged document copies.

Data Quality, Governance & Observability Framework

- Standardized automated controls for nulls, duplicates, referential integrity, schema drift, reconciliation, freshness, and business-rule validation, with failures routed through monitored orchestration and alerting paths.
- Combined cataloging, lineage, RBAC, audit logging, pipeline metrics, SLA monitoring, and structured operational runbooks to improve trust, traceability, and production support for critical data products.

Performance & Cost Optimization

- Applied Spark and warehouse optimization techniques including partition pruning, optimized joins, compaction, clustering, caching, workload right-sizing, and incremental processing to reduce avoidable compute and query scans.
- Established repeatable performance baselines and root-cause analysis practices for slow jobs, skewed workloads, data-volume anomalies, and failed pipelines before promoting changes to production.

Core Engineering Principles

- Design for restartability and idempotency so retries, backfills, replay, and late-arriving data do not corrupt downstream state.
- Treat data quality as part of delivery logic: validate schema, completeness, uniqueness, reconciliation, and freshness before publication.
- Prefer explicit data contracts, version-controlled transformations, automated tests, and controlled CI/CD promotion over manual fixes.
- Apply standard data-engineering controls to AI workloads: lineage, access control, quality, reproducibility, observability, and cost-aware orchestration.

CDC, Late Data & Backfill Strategy

- Used watermarking, business keys, source timestamps, MERGE logic, and idempotent write patterns to support incremental ingestion while preserving historical correctness.
- Designed replay and backfill paths that isolate late or corrected records, reprocess only affected partitions/windows, and avoid duplicate downstream state.

Dimensional Modeling & Analytics Serving

- Built fact/dimension models, conformed dimensions, surrogate keys, SCD Type 1/2 logic, and curated marts that separated reusable business entities from report-specific transformations.
- Aligned storage and serving patterns to workload needs—Parquet/Delta/Iceberg for scalable processing and Snowflake/warehouse marts for governed BI and analytical consumption.

Production Reliability & Incident Response

- Defined runbooks for failed dependencies, partial loads, schema changes, bad source files, SLA breaches, reprocessing, and production incident triage with clear recovery points.
- Converted recurring incidents into preventive engineering controls such as schema checks, freshness thresholds, reconciliation rules, alerts, test coverage, and automated recovery logic.

Governance & Security

- Applied least-privilege IAM/RBAC, managed identities, secrets management, encryption, audit trails, cataloging, and lineage across storage, compute, orchestration, and warehouse layers.
- Used data contracts and ownership metadata to make source expectations explicit and reduce downstream breakage when upstream structures or business rules changed.

Technical Leadership & Collaboration

- Led design reviews, code reviews, technical documentation, production troubleshooting, and knowledge sharing across data engineering, analytics, data science, QA, and platform teams.
- Translated business requirements into source-to-target designs, data models, SLAs, validation rules, and operational acceptance criteria that engineering and business stakeholders could jointly verify.

CERTIFICATIONS

- Databricks Certified Data Engineer Professional
- AWS Certified Data Engineer – Associate
- Microsoft Certified: Fabric Data Engineer Associate (DP-700)
- Snowflake SnowPro Core Certification (COF-C03)

EDUCATION

Master of Science in Information Studies - Trine University, Phoenix, Arizona

Bachelor of Technology in Computer Science and Engineering - Parul University, Vadodara, India