David Dallakyan

Data Engineer | Python · ETL · Data Quality · AWS · AI Training Data

New York, NY · US Citizen · daviddallakyan2005@gmail.com

linkedin.com/in/daviddallakyan2005 · github.com/daviddallakyan2005 · pypi.org/project/superset-toolkit

Summary

Python data engineer who owns production ETL end to end on Wirestock's AWS lakehouse, plus the pipelines that supply multimodal AI clients with licensed, rights-cleared media and high-quality embeddings: the real-world input that models train on. Designed, built, and operates 15+ ingestion pipelines and 100+ tested dbt/Trino models in clean, documented Python, with dbt tests and validation as the data-quality gate. Debugs and optimizes large-scale systems, clearing recurring Argo-controller and Trino OOM failures across ~2,000 production workflows and patching the MongoDB path of VectorDBBench while benchmarking 10+ vector systems for a 70M+ item search. Works independently as primary platform owner, aligning requirements with technical and non-technical stakeholders across teams, modeling and validating the data, monitoring production, and iterating, and built and structured the data team around that work. Member of the Trino organization. Ships production Python, SQL, TypeScript, Java, and Bash.

Experience

Wirestock · Stock media and AI datasets marketplace and A (Series A)

Cloud Data Engineer · January 2025 – Present · On site · Yerevan, Armenia

Primary owner of the AWS data platform. Independent ownership of production ETL, data quality, and the pipelines that deliver licensed media to multimodal AI clients.

ETL engineering in production Python

- Scaled company-wide ETL to 15+ production ingestion pipelines covering operational databases, MongoDB Atlas, partner feeds, and event sources landing in Iceberg tables served through Trino, by writing and operating clean, documented, reviewed Python on the AWS lakehouse (Spark, Iceberg, Nessie, Trino, dbt, DataHub, Argo on EKS).
- Researched, deployed, and now operate the entire lakehouse stack end to end, taking it from architecture trade study (Snowflake, BigQuery, Delta Lake; accepted ADRs) through production deployment on EKS, Terraform/CDK packaging, and daily operations.

- Migrated 5M+ media records from MySQL Aurora and legacy S3 to MongoDB Atlas, per-project S3, and REST APIs with zero downtime by building a resumable, idempotent Argo/Python runner and chained outbox events.
- Cut new-report delivery from 4–7 days to 4 hours by shipping 100+ tested dbt/Trino reporting pipelines with Kubernetes CI/CD, replacing person-dependent manual reporting.

Data quality, modeling & warehousing

- Made data quality a release gate rather than an afterthought: dbt tests, schema and freshness validation, and reconciliation checks run in CI/CD before any model or migration reaches production, with failures routed to owners.
- Modeled domain-owned Iceberg data products (dimensional and semantic models, federated access, DataHub catalog) so consuming teams discover and use governed data self-serve instead of queueing tickets.
- Automated Superset row-level security so access policy travels with the data products, keeping licensed and rights-sensitive assets governed across every downstream consumer.

Debugging & optimizing large-scale systems

- Restored production-pipeline reliability across ~2,000 Argo workflows, with zero Argo-controller and Trino OOM recurrences since rollout, by adding archival/retention policies, filtered event sources, and Iceberg metadata tuning, then documenting the practice in operational runbooks.
- Tuned throughput of bulk delivery with 48-way concurrent S3 transfers (s5cmd/boto3) and per-object failure tracking, turning multi-terabyte exports into a repeatable, monitorable job.

AI & analytics data products

- Delivered millions of licensed assets to multimodal AI and marketplace clients by building automated export pipelines: Trino/Iceberg selection, concurrent S3 transfer, and AWS CDK for production ML datasets, so clients receive documented, production-grade media as model input.
- Created a VDBMS-backed media search over 70M+ multimodal media by selecting the embedding model, producing embeddings through AWS Bedrock, and taking the system from evaluation to production on MongoDB Atlas.
- Owned that selection end to end: benchmarked 10+ vector systems (Qdrant, MongoDB Atlas Vector Search, OpenSearch k-NN, Pinecone, and others) on our own production datasets with Zilliz's VectorDBBench, patched and reported flaws found in its MongoDB benchmark, and paired the results with TCO and cost analysis plus direct evaluation calls with vendor sales and solutions engineers to negotiate pricing before choosing the production system.
- Built the internal reporting and analytics data products behind AI-lab client projects, translating business questions from non-technical stakeholders into modeled datasets, tested pipelines, and dashboards they operate themselves.

Ownership, collaboration & communication

- Led every data product end to end, from gathering and aligning requirements across teams and domains through architecture, release, and iteration, communicating trade-offs in writing to both technical and non-technical partners and coordinating Product, Design, Frontend, and Backend workstreams during Series A preparation.
- Built and structured the data team as it formed: defined ways of working, ran prioritization, and assigned tasks, while remaining hands-on as primary platform owner.
- Standardized written decision-making with accepted ADRs, runbooks, and documented data contracts, so the team can operate the platform without depending on any one person.

Open source

Member of the Trino organization; contributor to trinodb/trino (engine and Iceberg, Delta Lake, DuckDB, ClickHouse, Elasticsearch connectors). Also contributing in Apache Iceberg, DuckDB, dbt, Apache Superset, and Apache Spark. github.com/daviddallakyan2005

Own projects

- **superset-toolkit** — Published Python SDK on PyPI (GitHub Actions CI) to automate Superset charts, dashboards, datasets, and JWT auth. pypi.org/project/superset-toolkit
- **pdf-toolbox** — Desktop PDF/image tool (split, merge, preview).
- **armenian-ner-network** — Armenian named-entity recognition model (190 downloads in a month).
- **SwapMyClass** — Next.js + Supabase + Vercel cycle-matching for AUA class swaps.

Technical skills

- **Languages:** Python, SQL, NoSQL, TypeScript, C#, Java, Bash
- **ETL, modeling & quality:** Apache Spark/PySpark, Apache Iceberg, Nessie, Trino, dbt, Parquet/PyArrow, DataHub, data mesh, dimensional & semantic modeling, data warehousing, dbt tests & validation, ADRs/runbooks
- **AWS & platform:** EKS, S3, EventBridge, SQS, IAM/IRSA, Aurora, Bedrock, CloudWatch, Terraform, AWS CDK, Kubernetes, Docker
- **Orchestration:** Argo Workflows/Events, RabbitMQ outbox, Airflow (prototype DAGs)
- **Vector search & embeddings:** MongoDB Atlas Vector Search, OpenSearch k-NN, Qdrant, AWS Bedrock embeddings, VectorDBBench
- **Observability:** OpenTelemetry, Prometheus, Grafana, CloudWatch

Education

B.S. Computer Science, American University of Armenia · May 2026