

BASE SAS · EGP · DI STUDIO · SAS VIYA · DATAFLUX →
DATABRICKS LAKEHOUSE

Retire SAS. Run on the Lakehouse.

MigryX converts Base SAS programs, Enterprise Guide projects, DI Studio jobs, and SAS Viya workloads directly to **PySpark, Delta Lake, Unity Catalog, DLT pipelines, and Databricks Workflows** — with 95%+ parser accuracy, 99% with AI augmentation, and column-level lineage from day one.

99%AI
ACCURACY**60%+**COST
SAVINGS**10×**

FASTER

100%ON-PREM
CAPABLE[Run the Pilot Yourself →](#)[Email us →](#)

silver_claims.py · DLT

```
# Source: SAS DI Studio job
# Target: Databricks DLT (Silver)
import dlt
from pyspark.sql
    import functions as F

@dlt.table(
    name="silver_claims",
    comment="Cleansed claims"
)
@dlt.expect_or_drop(
    "valid_id", "id IS NOT NULL"
)
def silver_claims():
    return dlt.read("bronze_claims")
```

Conversion Accuracy 99.2%

Everything we ingest from the SAS ecosystem.

Purpose-built parsers for every corner of your SAS estate — Base SAS programs, Enterprise Guide projects, DI Studio jobs, SAS Viya/CAS workloads, and DataFlux quality jobs. Full fidelity, deterministic output, no approximation. Macro expansion, format resolution, and PROC translation included.



SAS Programs & Code

- Base SAS programs (.sas)
- DATA steps & SET / MERGE
- PROC SQL & pass-through
- PROC SORT / MEANS / FREQ / TRANSPOSE
- SAS Macros & %INCLUDE
- Formats, Informats & PROC FORMAT
- ODS output & report procedures
- SAS/IML & SAS/STAT



SAS Platform & Tools

- SAS DI Studio (ETL jobs)
- SAS Enterprise Guide (EGP)
- SAS Viya & CAS engine
- SAS Management Console
- Stored Processes & web services
- SAS Grid Manager & scheduling
- SAS DataFlux dfPower jobs



SAS Data Sources

- SAS datasets (.sas7bdat)
- Oracle, SQL Server, Teradata
- DB2, Netezza via SAS/ACCESS
- Flat files, CSV, Excel & XML
- ODBC & bulk-load connections
- External SAS libraries

SAS ARTIFACTS WE PARSE

.sas programs

.egp projects

DI Studio metadata

Stored Processes

Macro libraries

Format catalogs

CAS actions

SPDS schemas

JCL wrappers

Grid scripts

Autoexec.sas

DataFlux dfPower

The True Cost of Staying on SAS

SAS licenses, infrastructure, and tribal-knowledge risk compound annually. Manual rewrites take 3–5 years. Generic transpilers approximate macros, miss column-level lineage, and leave production at risk. MigryX delivers production Databricks code in weeks.

\$2M+

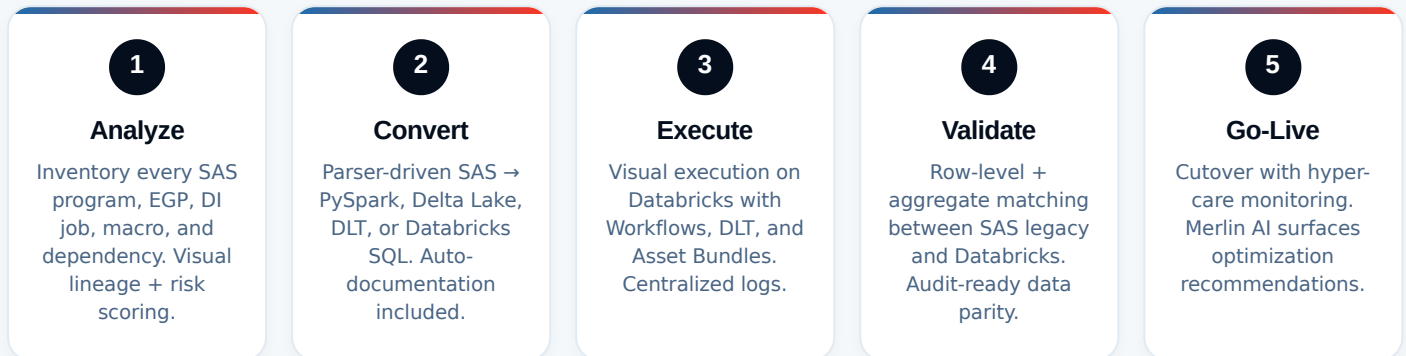
ANNUAL SAS SPEND

3–5yr

MANUAL REWRITE

One platform. Five lifecycle stages.

Every SAS-to-Databricks migration follows the same proven five-stage pipeline — from initial discovery through go-live and hyper-care. Each stage is fully automated, runs entirely in your environment, and produces audit-ready evidence.



1

Analyze

Inventory every SAS program, EGP, DI job, macro, and dependency. Visual lineage + risk scoring.

2

Convert

Parser-driven SAS → PySpark, Delta Lake, DLT, or Databricks SQL. Auto-documentation included.

3

Execute

Visual execution on Databricks with Workflows, DLT, and Asset Bundles. Centralized logs.

4

Validate

Row-level + aggregate matching between SAS legacy and Databricks. Audit-ready data parity.

5

Go-Live

Cutover with hyper-care monitoring. Merlin AI surfaces optimization recommendations.



Migration Planning · Compass

Compass scans every SAS file, scores it MIGRATE / ARCHIVE / DELETE based on usage, complexity, and recency. Wave-by-wave migration roadmap. Typical outcome: 60-80% data reduction, 5-10x analysis speed.



Code Conversion · Parser + Merlin AI

Custom AST parsers for SAS Base, Macro, PROC SQL, IML, EGP XML, and DI Studio metadata. 95%+ deterministic accuracy. Merlin AI resolves macro variables, runtime parameters, and ambiguous logic to push accuracy to 99%.



Visual Execution · Live on Databricks

One workspace shows visual lineage alongside live code with a direct Databricks session. See each step, the exact stop point, and full audit logs. Streamlines troubleshooting, cuts retesting, lowers compute costs.



Data Validation · Partitioned Matching

Row counts, common columns, mismatched columns, and aggregate metrics compared between SAS and Databricks outputs. Configurable keys, automatic schema checks, and exception trails give confidence to go live.



Deployment · Air-Gapped, In Your Network

Docker, Kubernetes, OpenShift, or Windows VM. Air-gapped capable — zero outbound connections. Source code never leaves your boundary. Deploy in under an hour. AWS, Azure, or GCP private clouds supported.

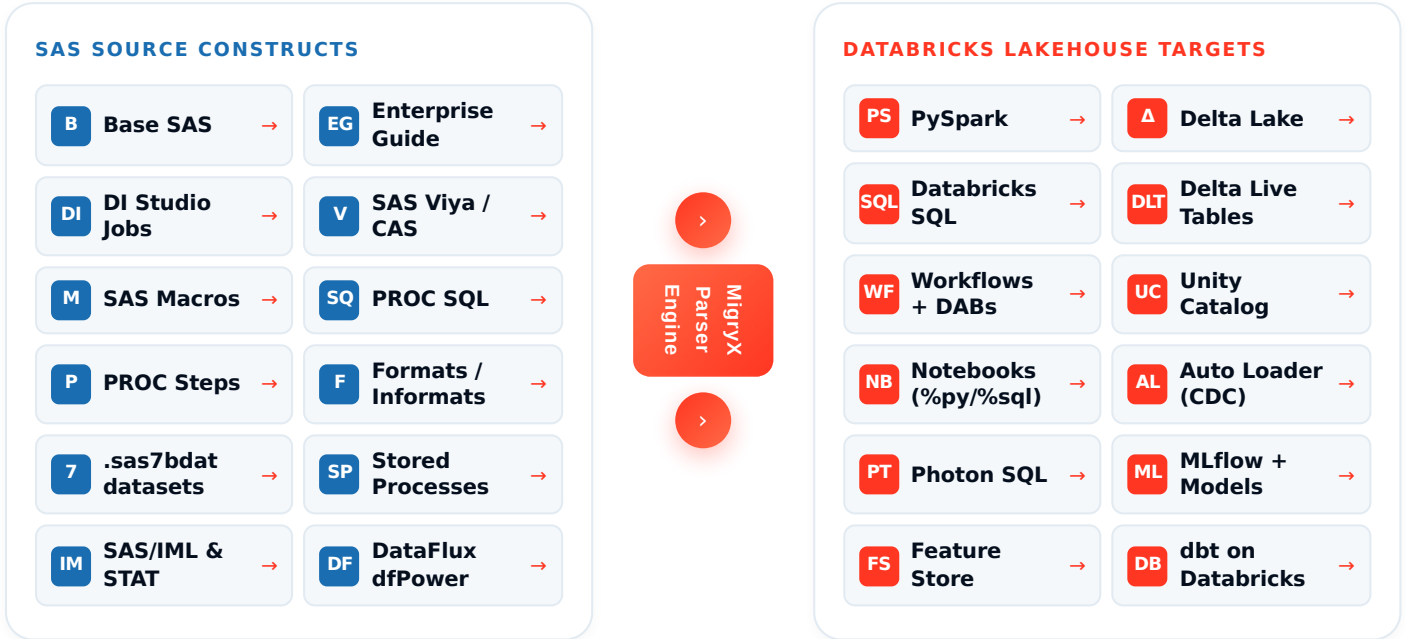


Orchestration · Workflows + DLT

Driver notebook or Databricks Job runner. Standardize on Delta and cloud storage. Schedule, monitor, auto-retry with centralized logs and metrics. Native integration with dbt, Airflow, and Git/CI.

Every SAS construct. One Databricks Lakehouse output.

Deterministic mapping from SAS source to Databricks-native targets. DATA steps become PySpark DataFrames. PROC SQL becomes Databricks SQL with Delta. Macros expand into Python functions. DI Studio jobs become Workflows with Asset Bundles. Every transformation auditable, every dependency preserved.



SAS CONSTRUCT	DATABRICKS OUTPUT	ACCURACY
DATA step · SET / MERGE	<code>spark.read.table()</code> · DataFrame ops	99%
PROC SQL · pass-through	Databricks SQL · ANSI compliant	99%
SAS Macros · <code>%macro</code>	Python functions · Jinja templates	95%+
PROC SORT/MEANS/FREQ	Spark <code>orderBy</code> / <code>agg</code> / <code>groupBy</code>	99%
DI Studio jobs · metadata	Databricks Workflows + Asset Bundles	95%+
.sas7bdat datasets	Delta tables · MERGE INTO + Z-ORDER	99%
SAS Viya · CAS actions	PySpark + Photon · MLflow models	95%+
PROC FORMAT · Informats	UDFs · cast / when expressions	99%

Run live on Databricks. Prove parity row by row.

Visual execution shows lineage and live code in one workspace with a direct Databricks session. Partitioned validation compares SAS legacy outputs against Databricks targets — row-level, column-level, aggregate. Audit-ready evidence packs ship with every run.

Visual Execution on Databricks

One view shows visual lineage alongside live code with a direct warehouse session. See each step, the exact stop point, and full upstream/downstream impact.

- ✓ Direct Databricks workspace session — live execution, not simulation
- ✓ Step-by-step visibility to any failure point with stack traces
- ✓ Visual Lineage shows upstream and downstream impact — retest only what matters
- ✓ Audit-ready logs and evidence packs auto-generated per run
- ✓ Native dbt, Airflow, Workflows, and Asset Bundle integration
- ✓ Centralized logs, retries, monitoring — production-grade orchestration

Data Validation SAS ↔ Databricks parity

Partitioned validation compares row-level and aggregate outputs between SAS legacy and Databricks targets using configurable keys and rules. Stakeholders trust the evidence.

- ✓ Row counts, common columns, mismatched columns, distinct values
- ✓ Configurable comparison keys, tolerances, and exception rules
- ✓ Automatic schema checks: types, lengths, nullability, formats
- ✓ Aggregate metrics: SUM, AVG, MIN, MAX, percentiles, distinct counts
- ✓ Side-by-side output comparison with mismatch drill-down
- ✓ Audit-ready exception trails — sign-off evidence for go-live

VALIDATION CHECK	SAS OUTPUT	DATABRICKS OUTPUT	STATUS
Row count · claims_silver	12,847,392	12,847,392	✓ MATCH
SUM(claim_amt)	\$847,293,182.44	\$847,293,182.44	✓ MATCH
Distinct policy_id	2,184,772	2,184,772	✓ MATCH
Schema · 47 columns	all types match	all types match	✓ PASS
AVG(claim_age_days)	14.273	14.273	✓ MATCH
NULL % · ssn_hash	0.012%	0.012%	✓ MATCH

Six modules. One self-service pilot.

Every module runs entirely inside your environment. No consultants, no vendor hand-holding. Deploy via Docker in under an hour, run the pilot on your own schedule.



Code Analysis

Assess thousands of SAS scripts. Map complexity, dependencies, and readiness. Clear scope, prioritized plan.



Visual Lineage

Visualize SAS code across jobs, datasets, and SQL. Speeds impact checks, lowers migration risk.



Code Conversion

Convert Base SAS, EGP, DI Studio, Viya into PySpark, Delta SQL, and DLT. Matched outputs.



Data Mapper

Auto-map .sas7bdat schemas to Delta tables. Naming conventions, data types, audit-ready visibility.



Auto Docs

Capture SAS source and Databricks target with parameters, dependencies, traceability — zero manual effort.



Data Matching

Compare source and target outputs at scale. Configurable keys, mismatch reports, audit trails.

Deployment · On-Premise · Air-Gapped Capable

No outbound connections. Zero external API calls. Source code and data stay entirely on your infrastructure.

DOCKER QUICK START · 8 CORES · 32 GB · 100 GB

VS CODE SERVER

port 1443

NGINX PROXY

port 3174

BACKEND API

port 3177

POSTGRESQL

port 5432

JUPYTER

port 8080

MERLIN

port 8000

```
docker run -d \
  -p 1443:1443 -p 3174:3174 \
  -p 3177:3177 -p 8080:8080 \
  --name migryx \
  -e PASSWORD="StrongPassword" \
  migryx:3.2.1
```

CLOUD · AWS · AZURE · GCP

AWS

EC2 m5.2xlarge · VPC · S3 · KMS · CloudWatch

Azure

D8s v3 · VNet · ADLS · Key Vault · Log Analytics

GCP

n2-standard-8 · VPC · GCS · Cloud KMS · Cloud Logging

K8s

Kubernetes / OpenShift on private cluster

Windows

VM with same hardware profile

Runs inside your VPC/VNet with private endpoints. Container images from your private registry. Secrets in your KMS. Data in S3/ADLS/GCS under your control. TLS with your certificates.

MIGRATION READINESS

Discovery & Insights

1 wk

100K

DURATION

LOC DISCOVERY

- ✓ Inventory workflows, macros, configs
- ✓ Visual lineage + risk assessment
- ✓ Self-service · runs in your env

FULL PILOT · END-TO-END

Convert · Execute · Validate

4–6 wk

10K

DURATION

LOC CONVERSION

- ✓ Discovery + pilot conversion
- ✓ Data matching + validation
- ✓ Enterprise data workflows

LARGE SCALE PILOT

Enterprise Migration

2–4 mo

100K

DURATION

LOC CONVERSION

- ✓ 1M LoC discovery scope
- ✓ Full project + JCL reports
- ✓ Production execution + cutover

Your competitors are already off SAS. Are you still maintaining .sas7bdat?

Every month on SAS compounds the debt — in licensing, talent scarcity, compliance risk, and missed AI/ML opportunity. The Databricks Lakehouse is the destination. MigryX is how you get there — at 10× speed.

⚡ **The SAS talent cliff is real:** The pool of engineers who know Base SAS, DI Studio, and Macro programming shrinks every year. The longer you wait, the more expensive your migration becomes — and the harder it is to find anyone who can do it. Modern data engineers want PySpark, Delta Lake, and MLflow.



Eliminate SAS Licensing

SAS licenses consume millions annually. Databricks delivers more capability for a fraction of the cost — typically 60%+ savings, with no per-CPU lock-in.



Unlock AI/ML on Your Data

SAS PROC code lands as MLflow experiments with Model Registry, Feature Store, and Model Serving. Production AI from day one — not a five-year project.



Streaming + Batch Unified

DLT pipelines unify streaming and batch with declarative `@dlt.table`. Auto Loader CDC ingestion, data quality expectations, enhanced autoscaling — out of the box.



Governance Built-In

Unity Catalog provides fine-grained access, attribute tags, and lineage from the moment your migrated tables land. Audit-ready GDPR, SOX, BCBS 239 reporting.



Preserve Decades of SAS Logic

Decades of SAS macros, formats, and PROC chains encode institutional knowledge. MigryX preserves every transformation with column-level lineage — nothing guessed.



Photon-Optimized Performance

Legacy SAS PROC SQL transpiled to Photon-optimized Databricks SQL on serverless SQL Warehouses with 500+ dialect function mappings. Faster queries, lower cost.

FOR DECISION MAKERS

See your SAS code land on Databricks — live, in 30 minutes.

No slides. No generic demos. We parse a sample of your actual SAS programs — Base SAS, EGP, DI Studio, or Viya — and show you converted PySpark with full Unity Catalog lineage and side-by-side validation. Self-service, in your environment, zero commitment.

60%+ COST SAVINGS **10×** FASTER **99%** AI ACCURACY **<1hr** TO DEPLOY

Run the Pilot Yourself →

Or send a sample — we'll convert your SAS to PySpark/DLT and return Unity Catalog lineage free.
hello@migryx.com · (617) 512-9530

MigryX  →  Databricks

Stop maintaining SAS. Start running on the Lakehouse.

Join the enterprises that have already landed millions of lines of Base SAS, EGP, DI Studio, and SAS Viya code on Databricks — PySpark-native, Unity Catalog-governed, MLflow-ready — in weeks, not years.

"MigryX cut our SAS-to-Databricks timeline from 3 years to 4 months. The Unity Catalog lineage alone justified the entire investment."



VP of Data Engineering

Fortune 500 Financial Services

"DI Studio jobs converted to PySpark + DLT in days. Side-by-side data matching gave our auditors complete confidence in cutover."



Chief Data Officer

Global Healthcare Enterprise

Free Proof of Concept — On Databricks



Send a sample of your SAS code — Base SAS, EGP, DI Studio, Viya, or DataFlux. We'll parse it, convert it, deploy to Databricks Workflows, and return Unity Catalog lineage — **free, no commitment, runs entirely in your environment.** [Start your POC →](#)

60%+

COST SAVINGS

10×

FASTER

99%

AI ACCURACY

Run the Pilot Yourself →

databricks.migryx.com

sas2py.com

hello@migryx.com

(617) 512-9530