

— BASE SAS · EGP · DI STUDIO · SAS VIYA · DATAFLUX → PYSARK

# Retire SAS. Run on distributed PySpark.

MigryX converts Base SAS, EGP, DI Studio, and SAS Viya workloads directly to **production-grade PySpark** — DataFrame APIs, Spark SQL, structured streaming, broadcast joins — runnable on Databricks, EMR, Dataproc, Fabric, Cloudera, or any Spark cluster. No vendor lock-in.

**99%**

AI ACCURACY

**60%+**

COST SAVINGS

**10×**

FASTER

**100%**

ON-PREM CAPABLE

## Everything we ingest from your SAS estate

### PROGRAMS & CODE

Base SAS (.sas)  
DATA · SET · MERGE  
PROC SQL · PROC steps  
SAS Macros · %INCLUDE  
Formats / Informats

### PLATFORM & TOOLS

DI Studio jobs  
Enterprise Guide (.egp)  
SAS Viya · CAS engine  
Stored Processes  
SAS Grid Manager

### DATA SOURCES

.sas7bdat datasets  
SAS/ACCESS connectors  
Oracle · SQL Server  
Teradata · DB2  
DataFlux dfPower



### 10× Faster Migration

Automated parser-driven conversion. Eliminate 3-5 years of manual rewrites.



### No Vendor Lock-In

Output runs on any Spark cluster. Move between Databricks, EMR, Dataproc freely.



### Merlin AI Augmentation

Resolves macros, runtime parameters, and ambiguous logic. Pushes accuracy to 99%.



### Audit-Ready Validation

Row-level + aggregate matching. Side-by-side parity proof for go-live confidence.

[Schedule a Demo →](#)[Email us →](#)

## TRANSLATION MAPPING

## Every SAS construct. Native PySpark output.

Deterministic mapping from SAS source to PySpark-native targets. Every transformation auditable, every dependency preserved, every macro expanded. 95%+ parser accuracy out of the box, 99% with Merlin AI augmentation.

SAS CONSTRUCT	PYSPARK OUTPUT	ACCURACY
DATA step · SET / MERGE	DataFrame.union() · join() · withColumn()	99%
PROC SQL · pass-through	Spark SQL · ANSI compliant	99%
SAS Macros · %macro	Python functions · Jinja2 templates	95%+
PROC SORT/MEANS/FREQ	orderBy · groupBy.agg · count	99%
DI Studio jobs · metadata	Airflow DAGs · driver notebooks	95%+
.sas7bdat datasets	Parquet / Delta / Iceberg tables	99%

## NATIVE PYSPARK OUTPUTS

## Production-ready PySpark from day one.

## COMPUTE &amp; LOGIC

Spark DataFrames

Spark SQL   PySpark UDFs

pandas UDFs

Structured Streaming

Broadcast Joins   AQE Hints

## STORAGE &amp; PIPELINES

Delta Lake   Iceberg Tables

Parquet / ORC   Hive tables

Auto Loader   MERGE INTO

## GOVERNANCE &amp; AI

Hive Metastore

AWS Glue Catalog

Unity Catalog   Apache Atlas

OpenLineage   MLflow

**Distributed Compute Native**

ETL logic converted to PySpark DataFrames running across any Spark cluster — Databricks, EMR, Dataproc, Fabric.

**Streaming + Batch Unified**

Structured Streaming for CDC and near-real-time. Same DataFrame API across both modes.

**Engine Portability**

No vendor lock-in. Output runs on any Spark cluster — self-managed, cloud-managed, or hybrid.

**Catalog-Agnostic Governance**

Lineage publishes to Hive Metastore, Unity Catalog, AWS Glue, or Apache Atlas — your choice.

5-STAGE METHODOLOGY

# From SAS estate to production PySpark — in five proven steps.

Every migration follows the same five-stage pipeline — fully automated, runs entirely in your environment, produces audit-ready evidence at each stage.

<b>1</b> <b>Analyze</b> Inventory SAS programs, EGP, DI jobs. Visual lineage + risk scoring.	<b>2</b> <b>Convert</b> Parser-driven SAS → PySpark translation. Auto-documentation included.	<b>3</b> <b>Execute</b> Visual execution on PySpark with centralized logs and orchestration.	<b>4</b> <b>Validate</b> Row-level + aggregate matching SAS legacy ↔ PySpark.	<b>5</b> <b>Go-Live</b> Cutover with hyper-care monitoring and Merlin AI optimization.
--	---	--	---	--

<b>99%</b> CONVERSION ACCURACY	<b>10×</b> FASTER THAN MANUAL	<b>0</b> VENDOR LOCK-IN	 SPARK-NATIVE
-----------------------------------	----------------------------------	----------------------------	---

RUN THE PILOT YOURSELF

<p><b>MIGRATION READINESS</b></p> <p>Discovery &amp; Insights</p> <p><b>1 wk</b>    <b>100K</b>  <small>DURATION    LOC DISCOVERY</small></p> <ul style="list-style-type: none"> <li>✓ Inventory + dependency mapping</li> <li>✓ Visual lineage + risk scoring</li> <li>✓ Self-service · runs in your env</li> </ul>	<p><b>FULL PILOT · END-TO-END</b></p> <p>Convert · Execute · Validate</p> <p><b>4–6 wk</b>    <b>10K</b>  <small>DURATION    LOC CONVERSION</small></p> <ul style="list-style-type: none"> <li>✓ Discovery + pilot conversion</li> <li>✓ Data matching + validation</li> <li>✓ Enterprise data workflows</li> </ul>	<p><b>LARGE SCALE PILOT</b></p> <p>Enterprise Migration</p> <p><b>2–4 mo</b>    <b>100K</b>  <small>DURATION    LOC CONVERSION</small></p> <ul style="list-style-type: none"> <li>✓ 1M LoC discovery scope</li> <li>✓ Full project + JCL reports</li> <li>✓ Production execution + cutover</li> </ul>
--	---	---

**Deployment · On-Premise · Air-Gapped**

No outbound connections. Zero external API calls. Source code and data stay entirely on your infrastructure. Deploy in under an hour.

<p><b>DOCKER · 8 CORES · 32 GB</b></p> <p>Single-command install on Linux or Windows VM. Self-service pilot — no consultants required.</p>	<p><b>CLOUD · AWS · AZURE · GCP</b></p> <p>Runs inside your VPC/VNet with private endpoints. Container images from your private registry.</p>	<p><b>K8S &amp; ENTERPRISE</b></p> <p>Kubernetes / OpenShift deployment. SOC 2 certified. Role-based access, full audit trails.</p>
--	---	---

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

No slides. No generic demos. Send a sample of your actual SAS code — Base SAS, EGP, DI Studio, or Viya — and we'll convert it, deploy it, and return column-level lineage. Free, no commitment, runs entirely in your environment.

[Schedule Your Demo →](#)