ProjectX

A SOVEREIGN, PORTABLE LAKEHOUSE · USE-CASE BRIEF

Keep your data in open tables, in your region, under your keys.

Tables in open Iceberg format on the storage you choose, encrypted with your own tenant keys, held in the regions your policy allows, governed at query time, and exportable whole when you leave.

The situation

The data platform is a vendor's format in a vendor's region. Residency is a promise in a contract, encryption keys are theirs, and leaving means a migration project nobody wants to fund.

How it works, step by step

01 **Store tables in an open format** · Lakehouse
Iceberg tables behind a standards-compatible REST catalog, on S3-compatible storage over MinIO or native buckets.

02 **Encrypt with your own keys** · Lakehouse
Data at rest is encrypted with per-tenant keys, and sensitive columns can be encrypted on their own.

03 **Keep it where it must stay** · the Data platform
Residency policies per tenant and classification decide which substrates data may live on, enforced at the decision point.

04 **Govern every query** · SQL & Analytics
Row and column rules are applied in the query plan, so the editor, dashboards, BI tools and the API return the same narrowed answer.

05 **Leave with everything** · Trust and security
A tenant export takes your data and configuration out with no lock-in.

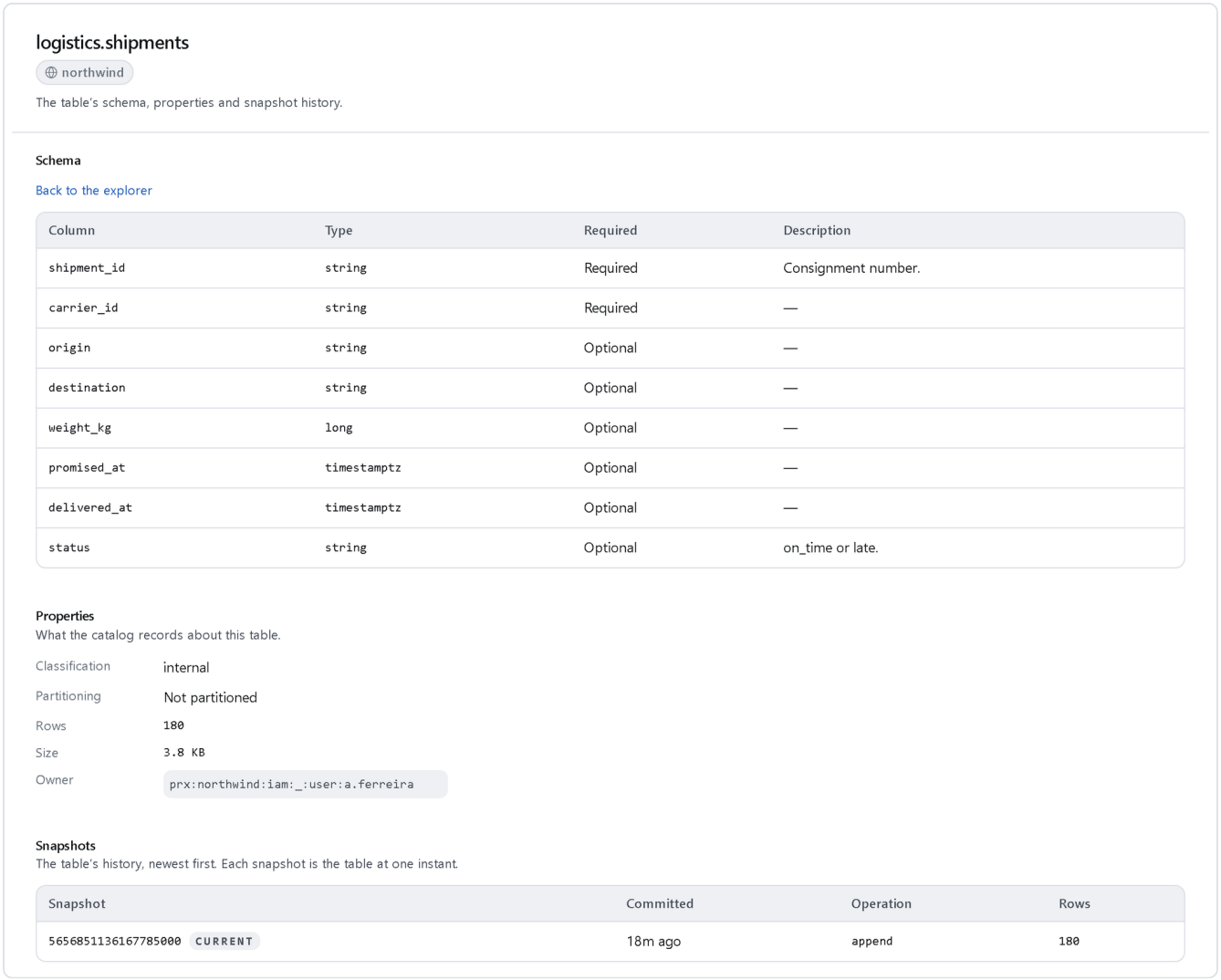
What you get

01
No proprietary format
Any engine that reads Iceberg can read your tables, and an export takes everything with you.

02
Residency enforced, not promised
Where data may live is a policy the platform checks, per tenant and classification.

03
History you can query
Time travel by timestamp or snapshot, and branches you can roll back.

In the console



One Iceberg table: schema, properties and the snapshot history. Screens shown with demo data.

Questions buyers ask

Can we bring our own storage?
Yes. The lakehouse stores tables on S3-compatible object storage, over MinIO or native buckets.

Who holds the keys?
Encryption at rest uses per-tenant keys, with column-level encryption for the most sensitive fields.