ProjectX

RECOVERING FROM A BAD CHANGE · USE-CASE BRIEF

Undo the bad change, whichever layer it was made in.

Roll a table back to the snapshot before the bad write, restore a database to a point in time, recover servers and objects from snapshots and versions, roll a workflow back to its last good version, and let reconciliation repair drift on its own.

The situation

A migration overwrote a table, a schema upgrade broke the database and an automation started doing the wrong thing, all on the same Friday. Each layer has its own backup story, and some have none.

How it works, step by step

01 **Roll the table back** · Lakehouse
Every write is a snapshot. Query the table as it was, then roll back or branch from the snapshot before the bad write.

02 **Restore the database** · Managed databases
Point-in-time recovery from automated backups, and version upgrades that roll back.

03 **Recover servers and files** · Backups
Scheduled server snapshots, object versioning and backup policies with retention tiers and cross-provider copies.

04 **Roll the automation back** · Workflows & Agents
Republish the last good version of a workflow; the history keeps every version that ran.

05 **Let drift repair itself** · Servers
Every resource has a desired state; drift is detected and alerted, and reconciliation brings it back.

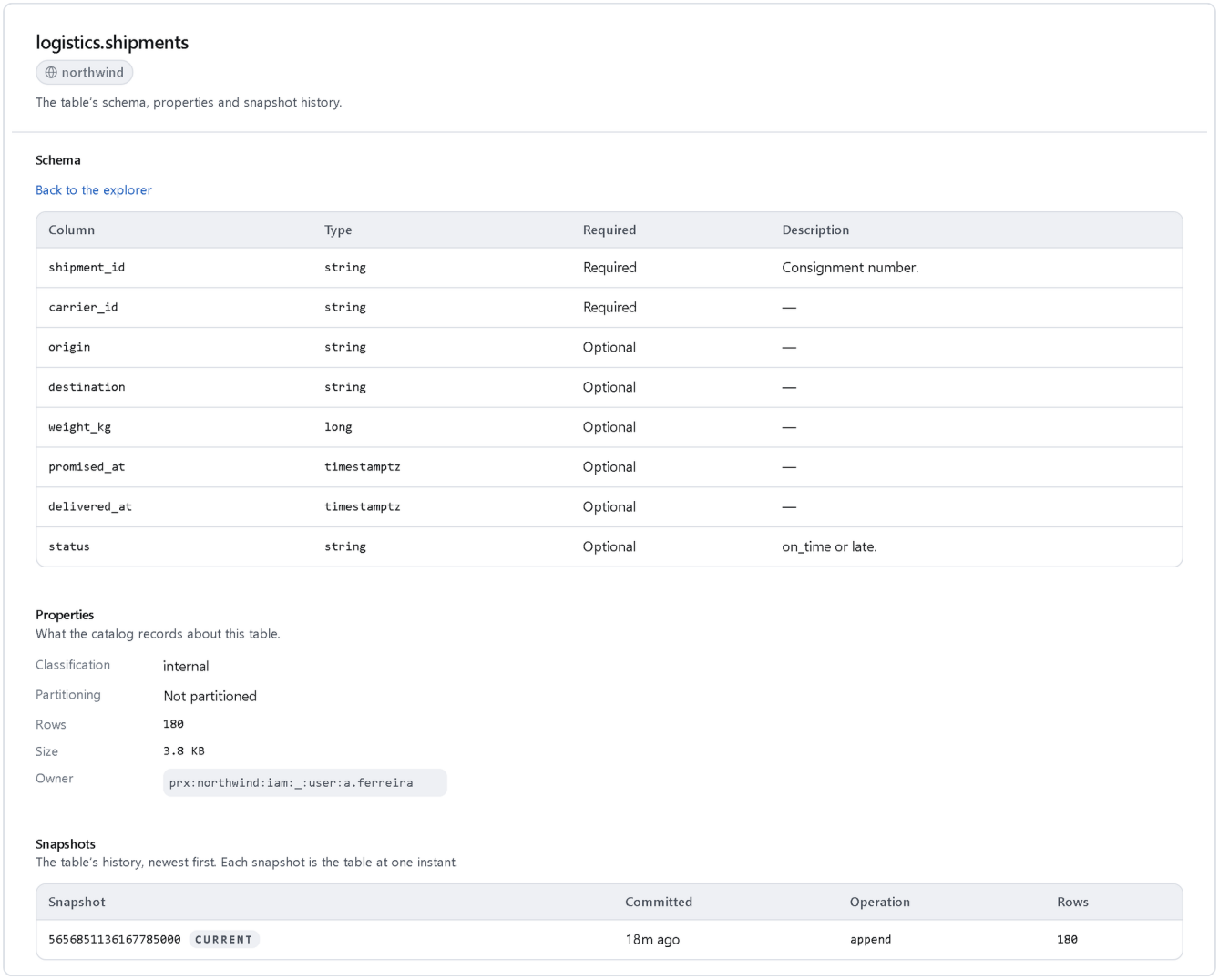
What you get

01
Every layer has an undo
Tables, databases, servers, objects and workflows each roll back on their own terms.

02
History, not just backups
Time travel and version history show what changed, not only a copy to restore.

03
Drift found, not discovered
Detected drift raises an alert instead of waiting for an outage.

In the console



logistics.shipments

northwind

The table's schema, properties and snapshot history.

Schema

Back to the explorer

Column	Type	Required	Description
shipment_id	string	Required	Consignment number.
carrier_id	string	Required	—
origin	string	Optional	—
destination	string	Optional	—
weight_kg	long	Optional	—
promised_at	timestamptz	Optional	—
delivered_at	timestamptz	Optional	—
status	string	Optional	on_time or late.

Properties

What the catalog records about this table.

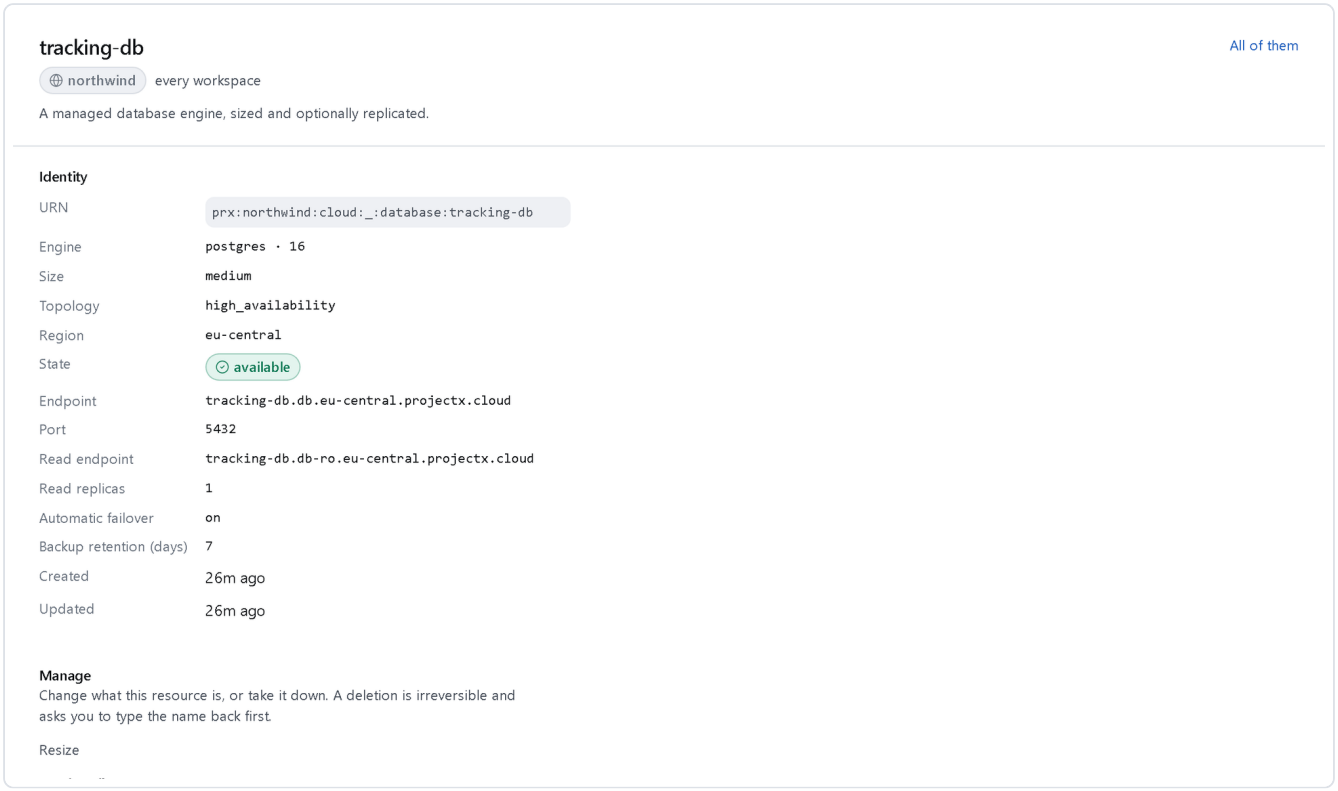
Classification	internal
Partitioning	Not partitioned
Rows	180
Size	3.8 KB
Owner	prx:northwind:iam:_:user:a.ferreira

Snapshots

The table's history, newest first. Each snapshot is the table at one instant.

Snapshot	Committed	Operation	Rows
5656851136167785000 CURRENT	18m ago	append	180

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



One database: primary and read endpoints, replicas, failover and backups. Screens shown with demo data.

Questions buyers ask

Can we rehearse a failover?

Yes. Managed databases support on-demand failover drills.

Are backups kept somewhere else?

Backup policies can copy to another provider, with retention tiers and an archive tier.