

PostgreSQL operational pain points and solutions

Monica Sarbu & Tudor Golubenco

September 2024

About us



Monica Sarbu
CEO @ Xata
@monicasarbu



Tudor Golubenco
CTO @ Xata
@tudor_g

xata.io

Xata - Postgres Database Platform

Fully managed Postgres service

- + Branching, location-independent databases
- + Zero downtime, reversible, schema migrations
- + Blue-green deployments, major version upgrades
- + CDC/Replication





Zero-downtime, reversible,
schema migrations for Postgres



Create Postgres extensions with
Zig



Postgres CDC with replication on DDL
changes

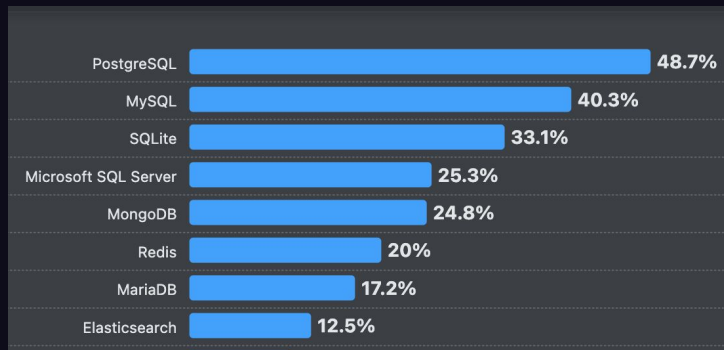
Company mission

To simplify the way developers work with data

Companies want to build applications faster, safer,
with less resources

What do developers want?

Postgres



(Why? Open source, open protocol, open ecosystem, many tools, etc.)

Yet, nothing is perfect

We **build tools** that solve Postgres' biggest **pain points** and make them part of the **Xata database platform.**

But which are the pain points?

We did our own research

Research

30+

In depth interviews
with industry leaders

50 -
10k

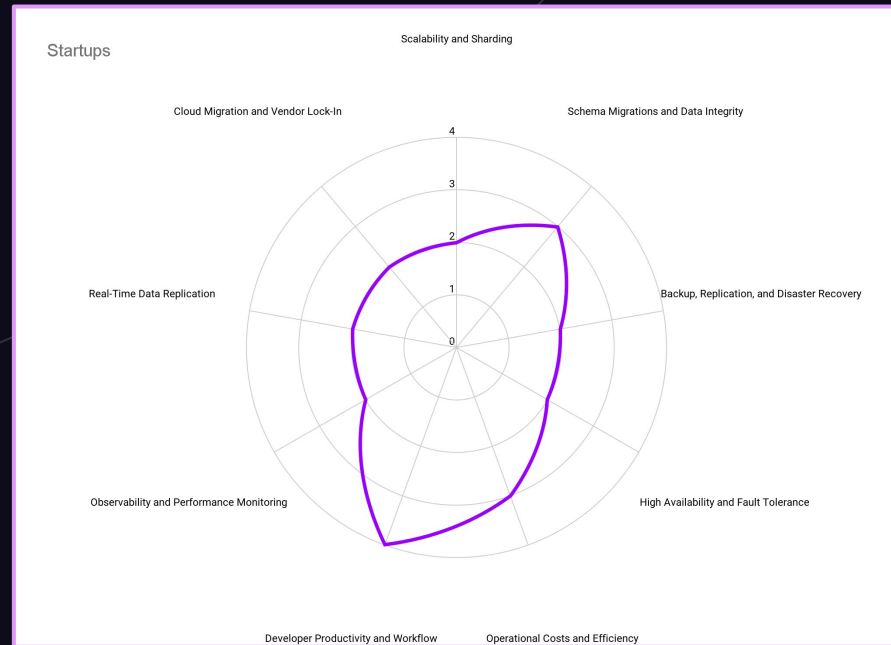
Employees in these
companies

10GB -
10PB

Scale of data and use
cases

Startups

- **Cost effective scalability:** Scale systems efficiently while minimizing infrastructure costs
- **Developer Productivity:** Streamline developer workflows to handle rapid iteration with lean teams.
- **Operational Costs:** Control costs while scaling infrastructure, avoiding over-provisioning.

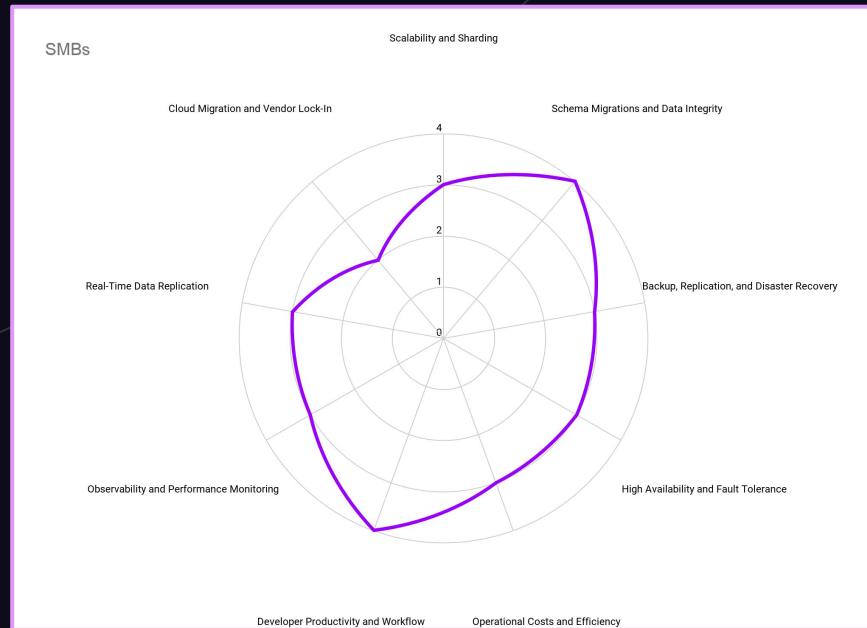


"Had a bunch of issues hitting a scale where the instance size wasn't good enough. Downtime was a problem."

"We needed faster iteration cycles with limited infrastructure. The team couldn't afford to spend extra time on database maintenance."

SMBs

- **Schema Migrations:** Ensure schema migrations are done without downtime.
- **Backup and Recovery:** Robust backup and disaster recovery systems as their data grows.
- **Operational Costs:** Balance performance and cost as they scale data systems.

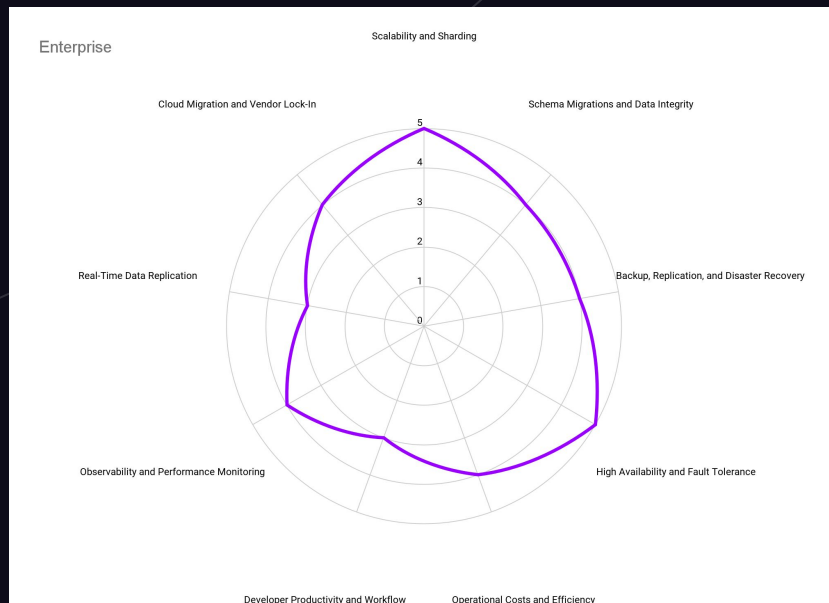


"Schema migrations have always been a challenge, especially without downtime. We don't have the luxury of waiting for off-hours."

"We handle over a terabyte of transactional data every day, and backups are critical. Ensuring recovery plans are solid is non-negotiable."

Enterprise

- **High Availability:** Zero downtime and fault tolerance across globally distributed systems.
- **Real-Time Data Replication:** Reliable real-time replication systems to manage large-scale, globally distributed data.
- **Performance Monitoring:** Require tools to optimize performance at scale and ensure reliable uptime.



"Real-time replication across multiple regions is our biggest challenge. We handle petabytes of data that need to be constantly synced."

"Vacuuming and scaling problems with Aurora are a constant headache. Would like something horizontally scalable like CockroachDB."

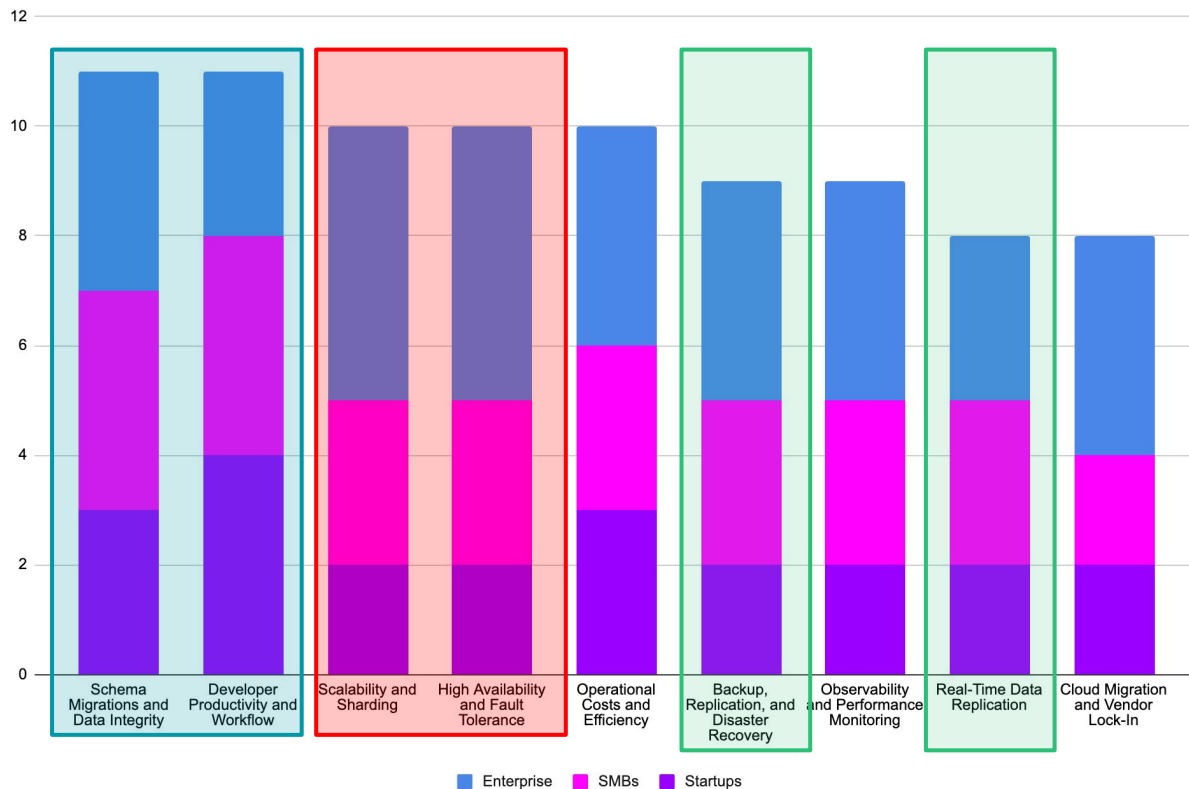
Takeaways

Deployment confidence and team developer experience

Replication to multiple regions and auxiliary data stores

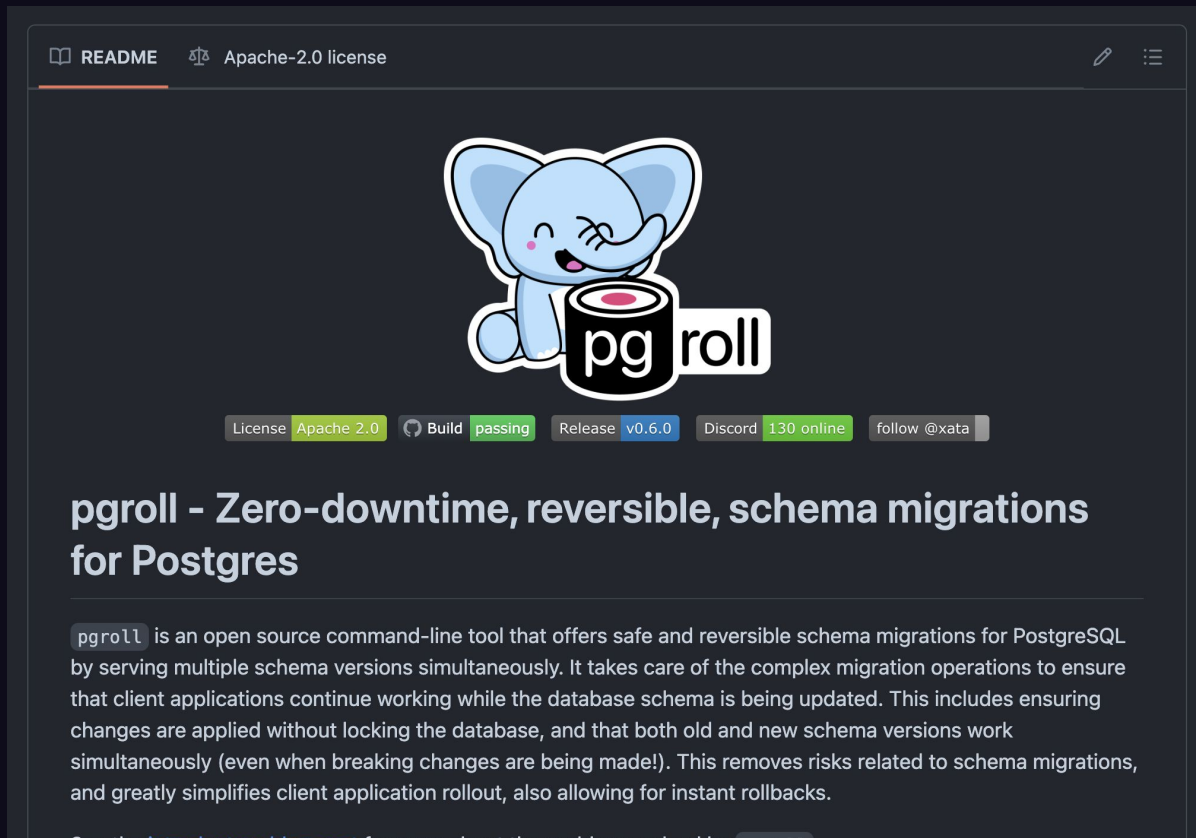
Sharding and global distribution of data

Pain points

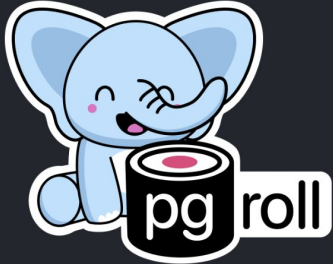


Schema changes

<https://github.com/xataio/pgroll>



README Apache-2.0 license



License Apache 2.0 Build passing Release v0.6.0 Discord 130 online follow @xata

pgroll - Zero-downtime, reversible, schema migrations for Postgres

`pgroll` is an open source command-line tool that offers safe and reversible schema migrations for PostgreSQL by serving multiple schema versions simultaneously. It takes care of the complex migration operations to ensure that client applications continue working while the database schema is being updated. This includes ensuring changes are applied without locking the database, and that both old and new schema versions work simultaneously (even when breaking changes are being made!). This removes risks related to schema migrations, and greatly simplifies client application rollout, also allowing for instant rollbacks.

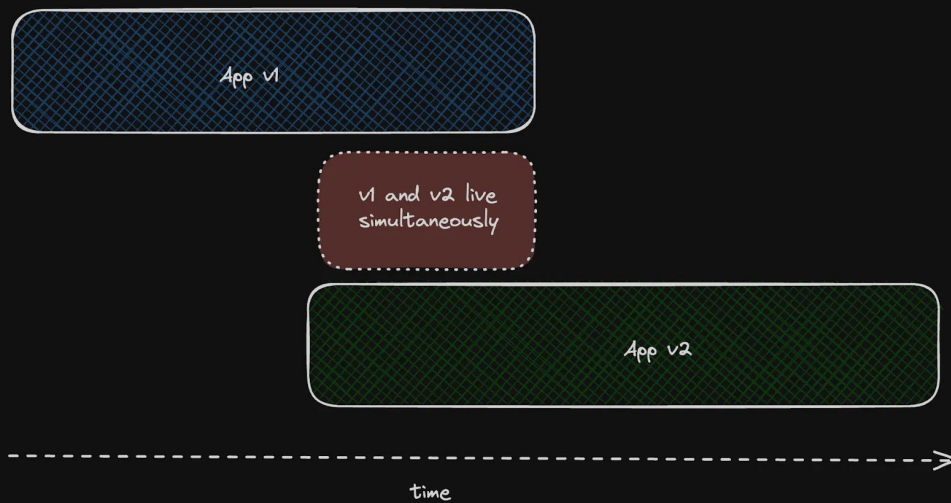
See the [introduction blog post](#) for more about the problems solved by `pgroll`.

Motivation

(Some) Postgres schema changes are difficult

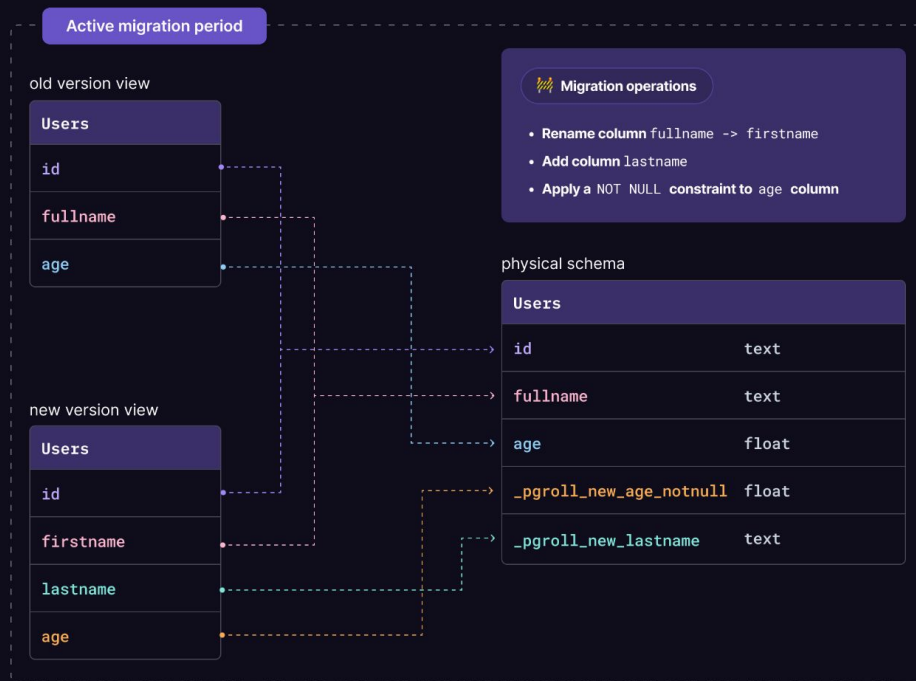
- Locking issues (e.g. on a busy table, in a fast a migration can wait at an exclusive lock and cause downtime)
<https://xata.io/blog/migrations-and-exclusive-locks>
- Constraints might require data back-filling (if you add a NOT NULL + UNIQUE column, how do you fill the data?)
- Backwards incompatible changes require a multiple step process (e.g. renames)

How



<https://xata.io/blog/multi-version-schema-migrations>

How



- Hidden columns are added to the physical table
- Data is backfilled and transformed in background
- Views hide or show the different columns

How - Application selects its version by setting the `search_path`

```
-- Switch back the new schema, which disallows `NULL`s in the `name` field
SET search_path TO mig_cq778qtl00e0bpredl0;

-- Attempt to insert a `NULL` value in the name field
INSERT INTO users(name) VALUES (NULL)
-- ERROR null value in column "name" of relation "users" violates not-null constraint

-- Switch back the old schema, which allows `NULL`s in the `name` field
SET search_path TO mig_cq778jdl00e0bpredk0;

-- Attempt to insert a `NULL` value in the name field
INSERT INTO users(name) VALUES (NULL)

-- Retrieve the data from the `users` table
SELECT * FROM users ORDER BY name DESC;
```

Benefits

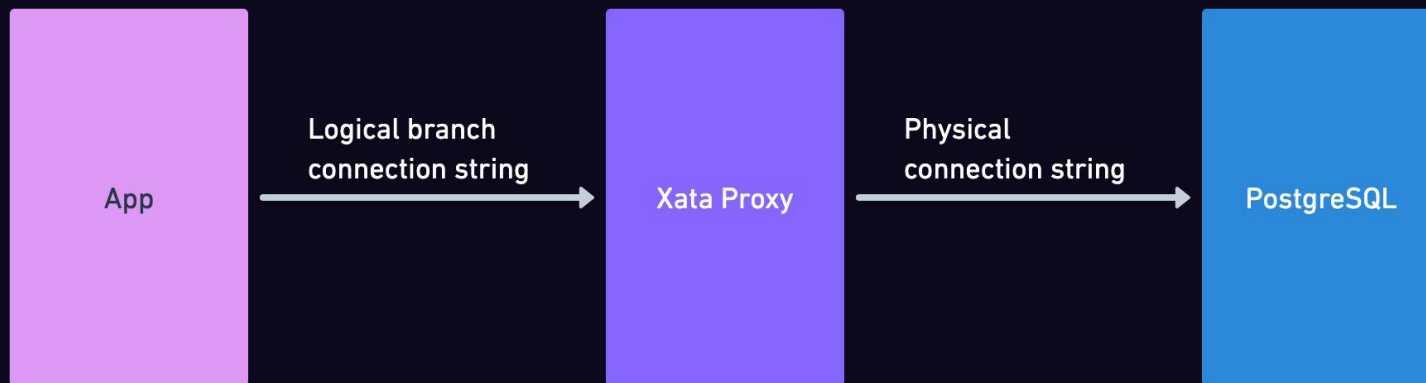
- Rollback is easy - just drop the views and intermediary columns.
- The tool takes care of locking issues and common issues.
- The merging workflow is always the same:
 - Start the pgoroll migration
 - Roll-out the application upgrade (can be blue-green)
 - Complete the pgoroll migration



Demo

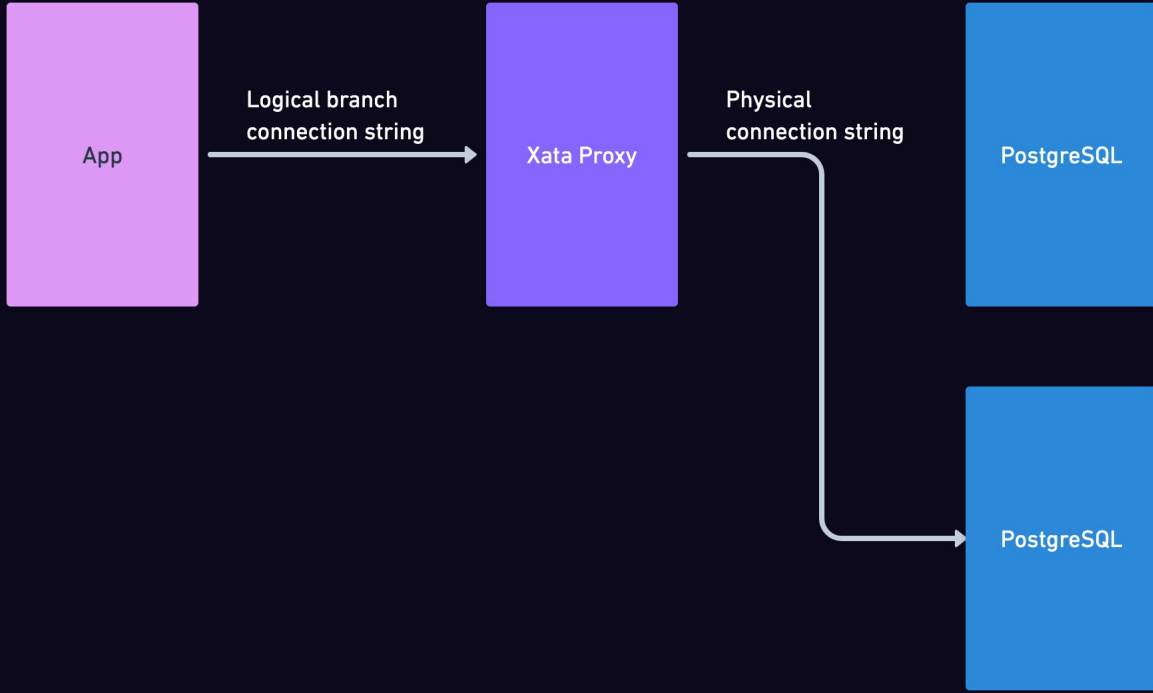
Location independent databases

Decoupled connection strings



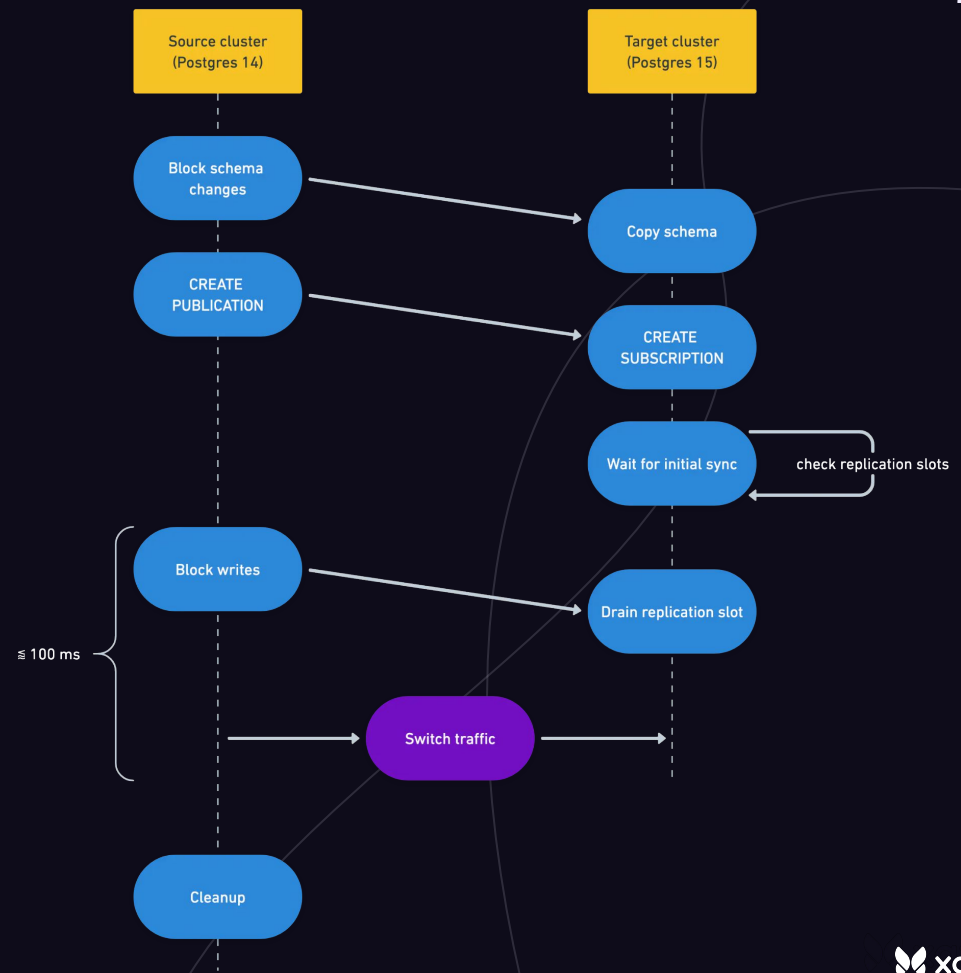
The connection string used by the application is decoupled from the location of the database.

This makes it possible to move the database without the application noticing



Moving branches without downtime

- Copy schema
- Create a logical replication replica
- Wait for the replica to catch up
- Block writes for a brief moment
- Switch traffic over (via the proxy)



Benefits of zero-downtime database moving

- Major version upgrades without downtime: simply move to an instance with a newer version
- Try different deployment options: auto-scaling vs provisioned, RDS vs Aurora






Demo

CDC / Replication to other data stores

📖 README 📄 Apache-2.0 license 📄 License ✎ ☰



License Apache 2.0 🔄 Build passing 🔗 reference Discord 114 online follow @xata

pgstream - Postgres replication with DDL changes

`pgstream` is an open source CDC command-line tool and library that offers Postgres replication support with DDL changes to any provided output.

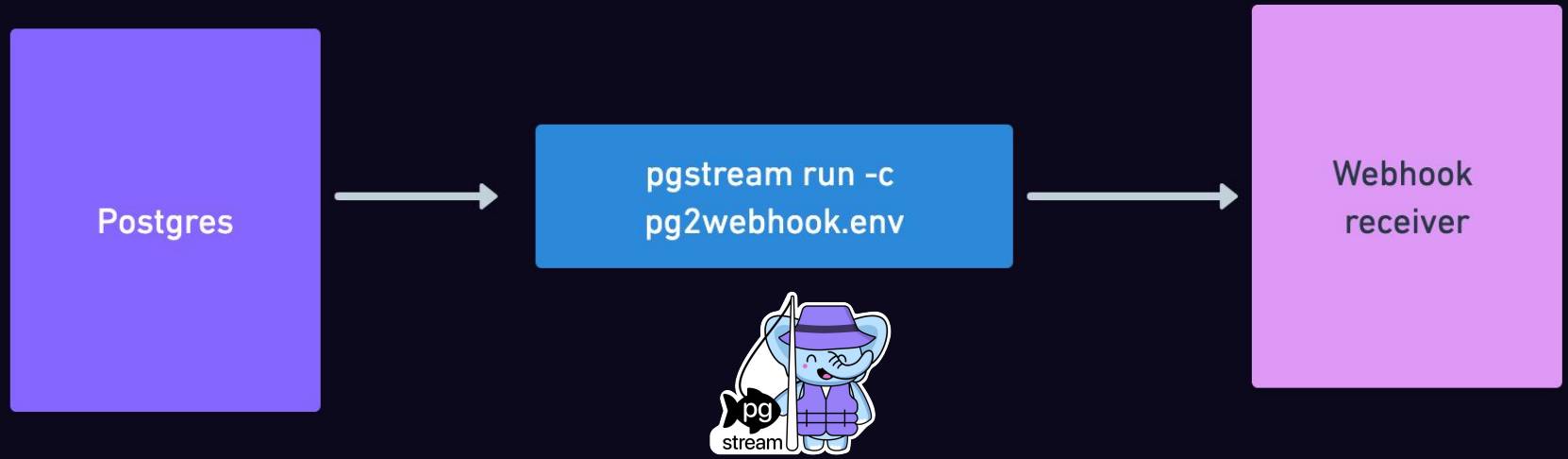
Features

Motivation

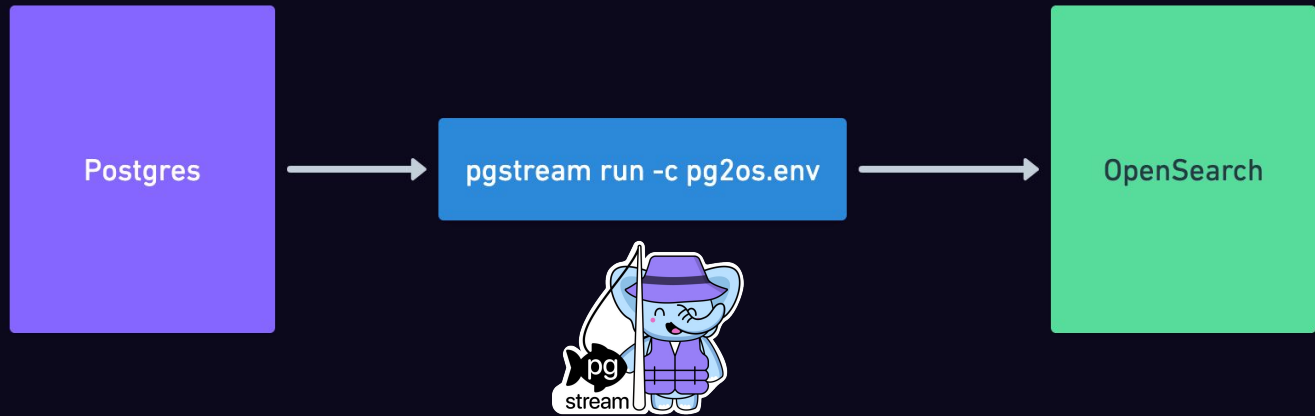
- Postgres logical replication doesn't include schema changes (only data changes)
 - Consumers have to guess types or query the schema out-of-band
- Automatically handle all the tricky parts of replication to OpenSearch
 - Batching for performance
 - Update in the presence of out-of-order events
 - Handle column renames
 - Handle of large values (TOAST)
- Get a nice JSON-stream of events or call webhooks.



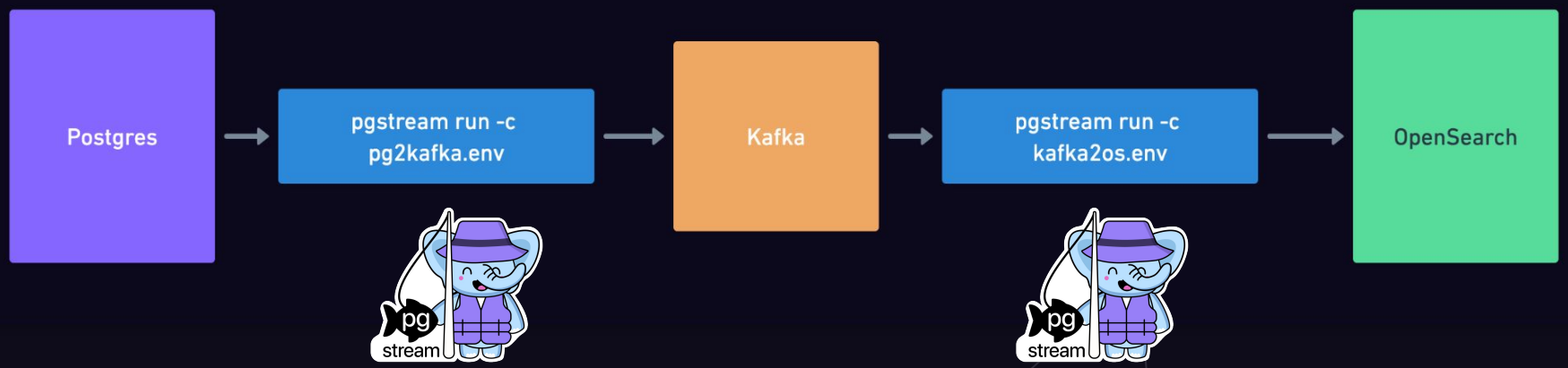
Webhooks for Postgres changes



Replication to Elasticsearch/OpenSearch



Replication to Elasticsearch/OpenSearch via Kafka





Demo



Q&A