PostgreSQL CDC with Go

Serhat Karabulut

@TrendyolGo - Localcommerce Promotion

github.com/3n0ugh

Table of Contents

Introduction

What is go-pq-cdc?

- A Change Data Capture (CDC) solution for efficiently replicating PostgreSQL data to other systems (e.g., Kafka, Elasticsearch) in real time. (RFC)

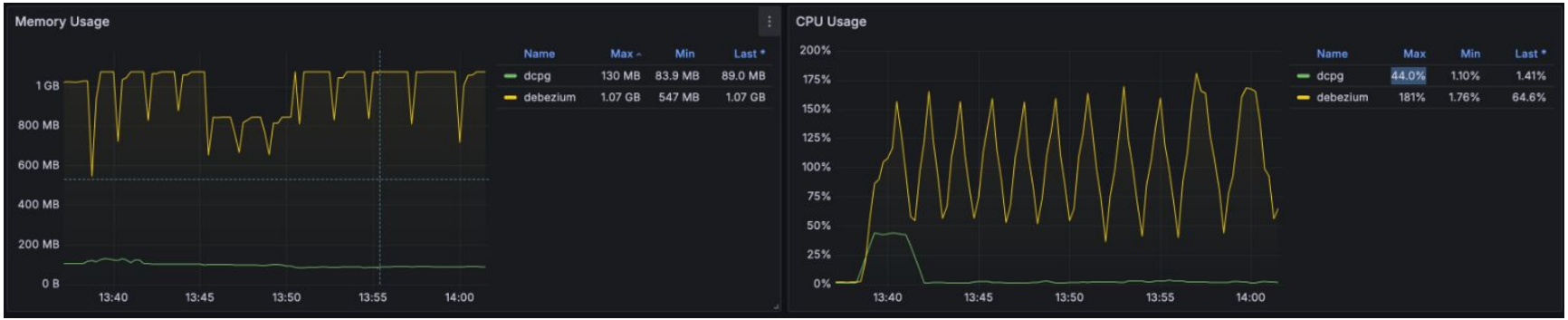
Purpose and Motivation

- Address limitations of existing solutions (e.g., Debezium) with a faster, lightweight alternative.
- Reduce PostgreSQL load for high-volume read scenarios by offloading reads to Elasticsearch.

Why Go?

Performance and Resource Efficiency

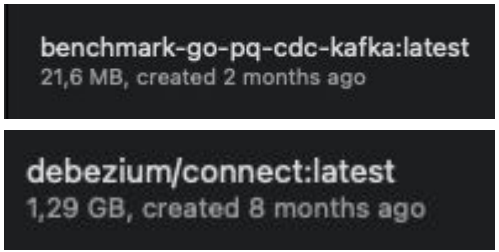
- Minimal memory and CPU usage compared to other CDC solutions.
- Fast startup and processing for real-time data handling.



Why Go?

Go Ecosystem Benefits

- Single binary deployment for ease of use in various environments (cloud-native, containerized systems, etc.).
- Go’s strong presence within Trendyol made it a natural choice, with the team already familiar with it.
- Ready-to-use components for Elasticsearch and Kafka integration via projects like [go-dcp]().



PostgreSQL Replication

Publication

A PostgreSQL object that defines the set of changes (INSERT, UPDATE, DELETE) from tables or schemas that are to be replicated.

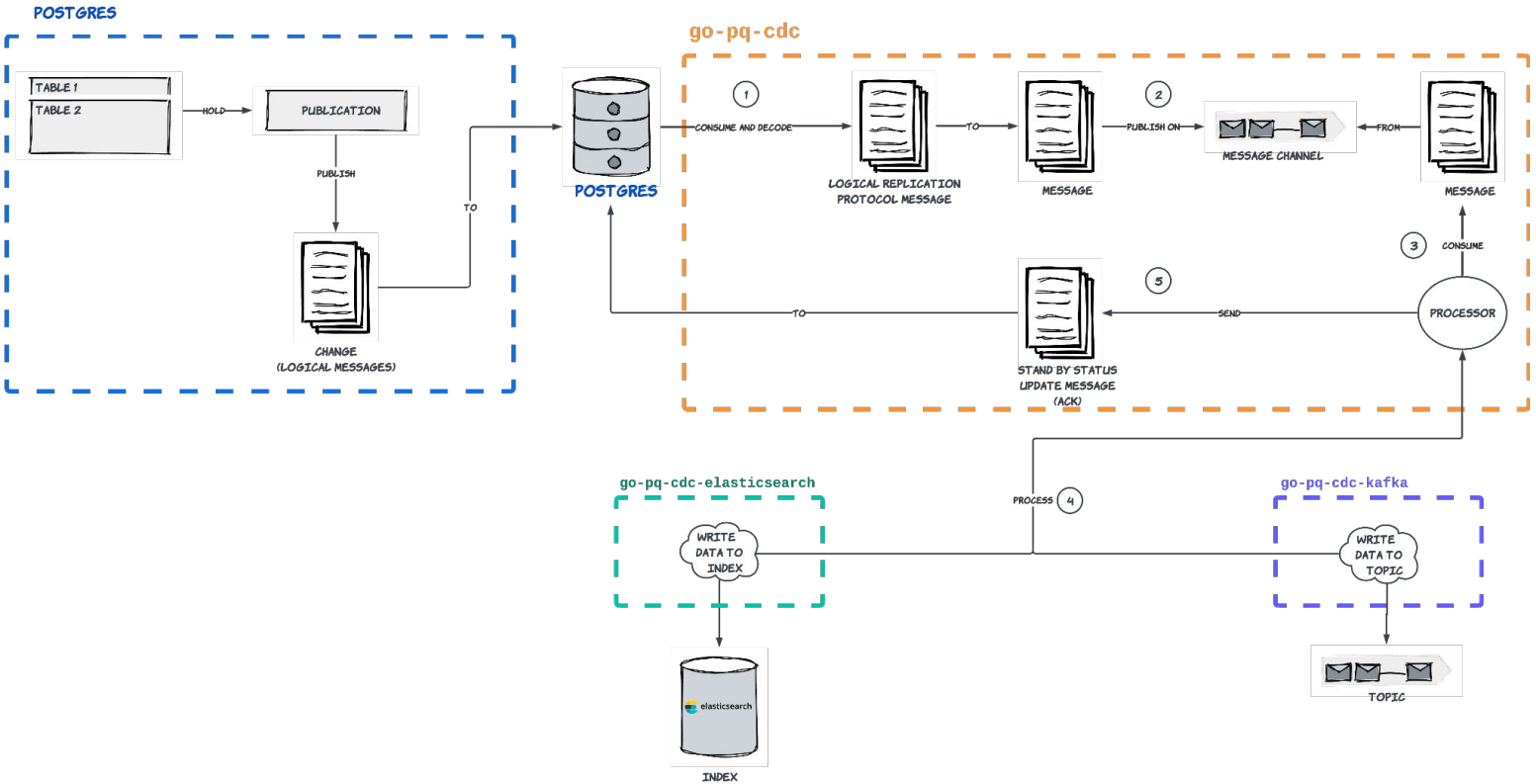
Replication Slot

A mechanism in PostgreSQL that ensures changes from the publication are retained until acknowledged by the subscriber (in this case, go-pq-cdc).

Logical Replication Messages

Data changes are sent as logical messages from the replication slot.

PostgreSQL Replication



How Go Was Used

PostgreSQL Replication with pgx and pglogrepl

- **pgx**: Lightweight PostgreSQL driver used for communication with the database.
- **pglogrepl**: Handles replication logic and ensures seamless streaming of changes.

Concurrency with Channels and Goroutines

- Efficient use of Go’s concurrency model.
- Channels manage communication between goroutines, ensuring high-throughput data replication without bottlenecks.

Code Insights

Thank You,
Any Questions