**PostgreSQL to MySQL CDC Pipeline Implementation**

**Overview**

I successfully implemented a Change Data Capture (CDC) pipeline to synchronize data between PostgreSQL (source) and MySQL (target) databases using Debezium and Apache Kafka. This documentation captures the complete working solution.

**System Architecture**

The implemented solution uses:

* PostgreSQL 16 as the source database
* MySQL as the target database
* Apache Kafka 4.0.0 in KRaft mode (ZooKeeper-less)
* Debezium 2.6.1.Final connectors

**Implementation Steps**

**1. PostgreSQL Configuration**

First, I configured PostgreSQL for logical replication:

***ALTER SYSTEM SET wal\_level = 'logical';***

***ALTER SYSTEM SET max\_replication\_slots = 10;***

***ALTER SYSTEM SET max\_wal\_senders = 10;***

Created the source database and user with appropriate permissions:

***CREATE DATABASE source\_db;***

***CREATE USER cdc\_user WITH PASSWORD 'cdc\_password';***

***ALTER ROLE cdc\_user WITH REPLICATION;***

***GRANT ALL PRIVILEGES ON DATABASE source\_db TO cdc\_user;***

**2. MySQL Configuration**

Set up the target database in MySQL:

***CREATE DATABASE target\_db;***

***CREATE USER 'cdc\_user'@'localhost' IDENTIFIED BY 'cdc\_password';***

***GRANT ALL PRIVILEGES ON target\_db.\* TO 'cdc\_user'@'localhost';***

**3. Kafka Setup**

Installed Kafka 4.0.0 in KRaft mode with the following configuration:

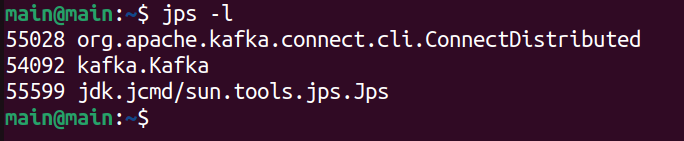
properties

# config/kraft/server.properties

* process.roles=broker,controller
* node.id=1
* controller.quorum.voters=1@localhost:9093
* listeners=PLAINTEXT://:9092,CONTROLLER://:9093
* log.dirs=/tmp/kafka-logs

Formatted the storage and started Kafka:

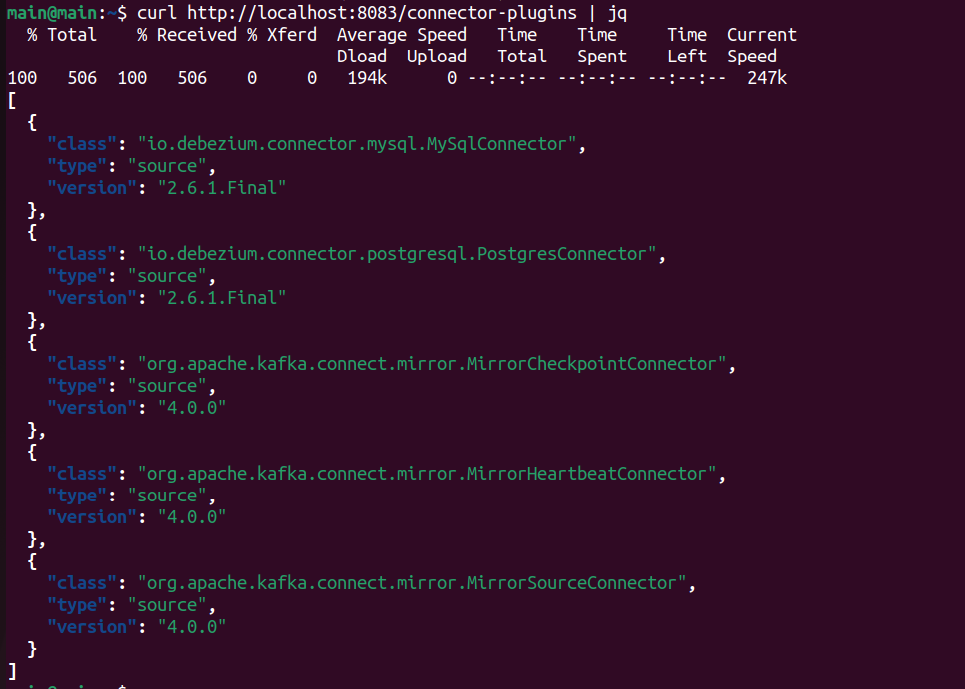
* /opt/kafka/bin/kafka-storage.sh format -t $UUID -c /opt/kafka/config/kraft/server.properties
* /opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/kraft/server.properties



**4. Debezium Connectors Installation**

Downloaded and installed the required connectors:

* ***wget https://repo1.maven.org/maven2/io/debezium/debezium-connector-postgres/2.6.1.Final/debezium-connector-postgres-2.6.1.Final-plugin.tar.gz***
* ***wget*** [***https://repo1.maven.org/maven2/io/debezium/debezium-connector-jdbc/2.6.1.Final/debezium-connector-jdbc-2.6.1.Final-plugin.tar.gz***](https://repo1.maven.org/maven2/io/debezium/debezium-connector-jdbc/2.6.1.Final/debezium-connector-jdbc-2.6.1.Final-plugin.tar.gz)



**5. PostgreSQL Source Connector Configuration**

After resolving permission issues, the final working configuration:

***{***

***"name": "postgres-source-connector",***

***"config": {***

***"connector.class": "io.debezium.connector.postgresql.PostgresConnector",***

***"database.hostname": "localhost",***

***"database.port": "5432",***

***"database.user": "cdc\_user",***

***"database.password": "cdc\_password",***

***"database.dbname": "source\_db",***

***"database.server.name": "pg\_server",***

***"topic.prefix": "pg\_server",***

***"table.include.list": "public.users",***

***"plugin.name": "pgoutput",***

***"publication.name": "dbz\_publication",***

***"publication.autocreate.mode": "disabled",***

***"slot.name": "debezium\_slot",***

***"tombstones.on.delete": "true"***

***}***

***}***

**6. MySQL Sink Connector Configuration**

The working JDBC sink connector configuration:

***{***

***"name": "mysql-sink-connector",***

***"config": {***

***"connector.class": "io.debezium.connector.jdbc.JdbcSinkConnector",***

***"tasks.max": "1",***

***"connection.url": "jdbc:mysql://localhost:3306/target\_db?useSSL=false",***

***"connection.username": "cdc\_user",***

***"connection.password": "cdc\_password",***

***"topics": "pg\_server.public.users",***

***"table.name.format": "users",***

***"insert.mode": "upsert",***

***"primary.key.mode": "record\_key",***

***"primary.key.fields": "id",***

***"auto.create": "false",***

***"auto.evolve": "false",***

***"delete.enabled": "true"***

***}***

***}***

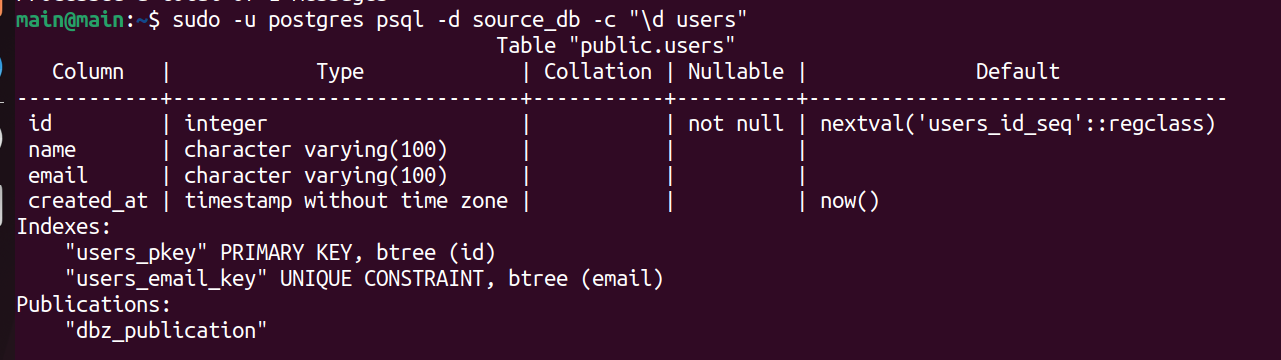


**Verification Process**

**Successful Data Flow**

Inserted test records in PostgreSQL:

***INSERT INTO users (name, email) VALUES ('Test User', 'test@example.com');***



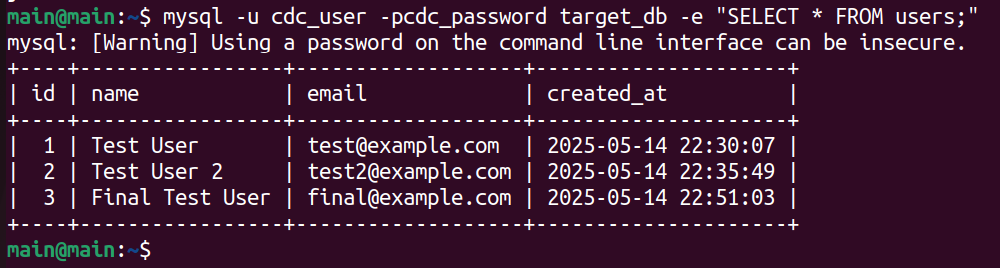
Verified the records appeared in Kafka:

***/opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic pg\_server.public.users --from-beginning***



Confirmed data was replicated to MySQL:

***mysql -u cdc\_user -pcdc\_password target\_db -e "SELECT \* FROM users;"***



**Troubleshooting Highlights**

**Resolved Issues**

PostgreSQL Permissions:

* Had to grant table ownership to cdc\_user
* Manually created the publication

Kafka KRaft Mode:

* Required proper storage formatting
* Needed correct controller configuration

Sink Connector Configuration:

* Needed to specify primary key fields explicitly
* Required proper topic naming convention

**Conclusion**

The implemented CDC pipeline successfully maintains near-real-time synchronization between PostgreSQL and MySQL. The solution handles inserts, updates, and deletes with minimal latency .