PostgreSQL 16 + Debezium CDC Setup Guide

PostgreSQL 16 + Debezium CDC Setup Guide

PostgreSQL 16 + Debezium CDC Setup Guide (Step-by-Step)

This document explains:

- Enabling PostgreSQL password login
- Creating database + users
- Setting replication permissions
- Configuring PostgreSQL for Debezium
- Installing Debezium PostgreSQL Kafka plugin
- Creating Kafka connector

---

1. Enable Password Authentication in PostgreSQL

Edit:

```
sudo nano /var/lib/pgsql/16/data/pg_hba.conf
```

Change:

```
local   all   postgres   peer
local   all   all        peer
```

To:

```
local   all   postgres   md5
local   all   all        md5
```

Save and exit.

---

2. Restart PostgreSQL

```
sudo systemctl restart postgresql-16
```

---

3. Set Password for postgres

```
sudo -i -u postgres
psql
```

Inside psql:

```
ALTER USER postgres WITH PASSWORD 'StrongPass123!';
```

Exit:

```
\q
exit
```

---

4. Create New Database

```
psql -U postgres -W
```

```
CREATE DATABASE mydb;
```

---

5. Create Debezium User

```
CREATE USER debezium WITH PASSWORD 'DebeziumPass123!';
ALTER ROLE debezium REPLICATION LOGIN;
```

6. Grant User Permissions

```
GRANT CONNECT ON DATABASE mydb TO debezium;
\c mydb
GRANT USAGE ON SCHEMA public TO debezium;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO debezium;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO debezium;
```

7. Create Publication

```
CREATE PUBLICATION dbz_publication FOR ALL TABLES;
```

8. Configure PostgreSQL for Logical Replication

Edit:

```
sudo nano /var/lib/pgsql/16/data/postgresql.conf
```

Set:

```
wal_level = logical
max_wal_senders = 10
max_replication_slots = 10
shared_preload_libraries = 'pgoutput'
```

Restart:

```
sudo systemctl restart postgresql-16
```

9. Update pg_hba.conf for Debezium Access

```
sudo nano /var/lib/pgsql/16/data/pg_hba.conf
```

Add:

```
host    replication    debezium     0.0.0.0/0     md5
host    all            debezium     0.0.0.0/0     md5
```

Restart:

```
sudo systemctl restart postgresql-16
```

---

10. Install Debezium PostgreSQL Connector Plugin (Kafka)

```
cd /opt/kafka/connector/
sudo curl -O https://repo1.maven.org/maven2/io/debezium/debezium-connector-
postgresql/3.3.0.Final/debezium-connector-postgresql-3.3.0.Final-plugin.tar.gz
sudo tar -xvf debezium-connector-postgresql-3.3.0.Final-plugin.tar.gz
```

Restart Kafka Connect:

```
sudo systemctl restart kafka-connect
```

---

11. Verify Plugin

```
curl localhost:8083/connector-plugins | jq
```

Expected:

```
io.debezium.connector.postgresql.PostgresConnector
```

---

12. Create Kafka Connector JSON

File: `postgres-connector.json`

```
{
  "name": "postgres-connector",
  "config": {
    "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
    "tasks.max": "1",
    "database.hostname": "127.0.0.1",
    "database.port": "5432",
    "database.user": "debezium",
    "database.password": "DebeziumPass123!",
    "database.dbname": "mydb",
    "topic.prefix": "pg",
    "slot.name": "debezium_slot",
    "publication.name": "dbz_publication",
    "plugin.name": "pgoutput",
    "table.include.list": "public.*",
    "tombstones.on.delete": "false"
  }
}
```

---

13. Register Connector

```
curl -X POST -H "Content-Type: application/json"      --data @postgres-connector.json
http://localhost:8083/connectors
```

---

14. Check Status

```
curl http://localhost:8083/connectors/postgres-connector/status | jq
```

? Done

Your PostgreSQL → Kafka Debezium CDC pipeline is now fully configured.

Create table and test

Connect to PostgreSQL

If your DB name is `mydb`:

```
psql -U postgres -W -d mydb
```

OR using the Debezium user (read-only but good for testing SELECT):

```
psql -U debezium -W -d mydb
```

? 2. Create a table

Inside psql:

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(150),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Debezium will track this table because:

- You created a publication with `FOR ALL TABLES`
- Debezium captures all changes in the `public` schema

? 3. Insert sample data

```
INSERT INTO users (name, email)
VALUES
('Golam Kibria', 'gkibria@example.com'),
('John Doe', 'john@example.com'),
('Alice Rahman', 'alice@example.com');
```

? 4. Query table to confirm

```
SELECT * FROM users;
```

You should see:

```
 id |     name      |        email         |      created_at
----+---------------+----------------------+-------------------------
 1  | Golam Kibria  | gkibria@example.com | 2025-11-30 23:50:12
 2  | John Doe      | john@example.com    | 2025-11-30 23:50:12
 3  | Alice Rahman  | alice@example.com   | 2025-11-30 23:50:12
```

? 5. Verify Debezium CDC events (optional)

In Kafka, list topics:

```
/opt/kafka/bin/kafka-topics.sh --bootstrap-server localhost:9092 --list
```

Expected:

```
pg.public.users
pg.public.users-value
pg.public.users-key
```

Consume the topic:

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