**Tạo kafka cluster và replication**

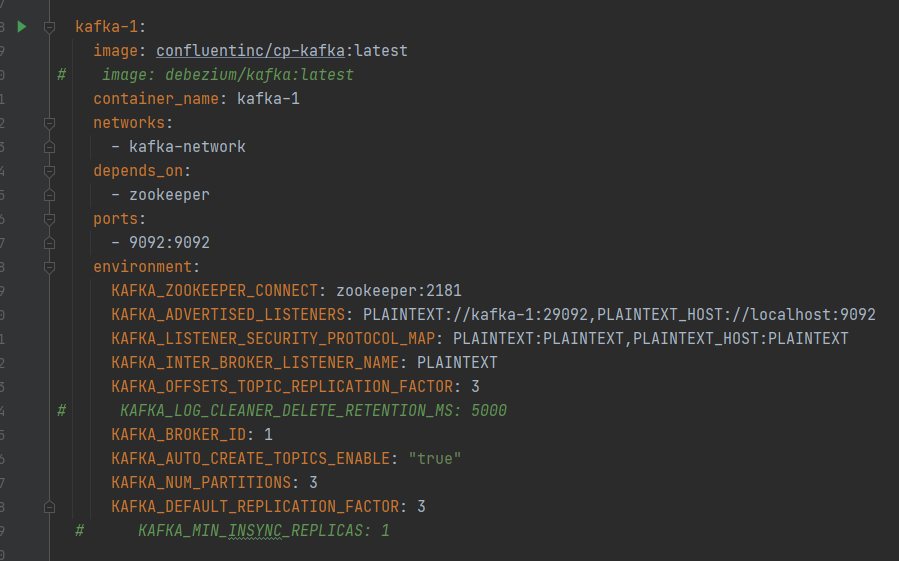
**B1: file docker-compose.yml:**

Cấu hình Zookeeper: quản lý kafka cluster

Cấu hình các 3 kafka có 3 partition và replication =3;

KAFKA\_NUM\_PARTITIONS: 3

KAFKA\_DEFAULT\_REPLICATION\_FACTOR: 3



Hoặc cũng có thể cấu hình ở file connect debezium

"topic.creation.default.replication.factor": 3,

"topic.creation.default.partitions": 3



**1: database.\*** cấu hình là các tham số kết nối cho cơ sở dữ liệu postgres

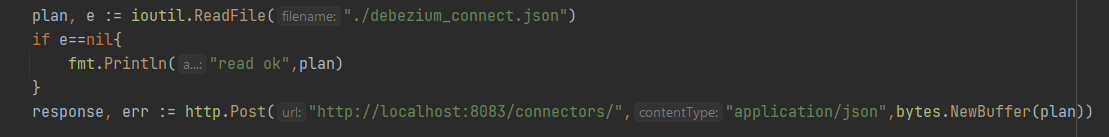
**2: database.server.name** là tên tự gán cho cơ sở dữ liệu của mình

**3: table.whitelist** là trường thông báo cho trình kết nối debezium chỉ đọc các thay đổi dữ liệu từ bảng đó. Tương tự, cũng có thể đưa các bảng hoặc lược đồ vào danh sách trắng hoặc danh sách đen. Theo mặc định, debezium đọc từ tất cả các bảng trong một lược đồ.

**4: connector.class** là trình kết nối được sử dụng để kết nối với cơ sở dữ liệu postgres

**5: name** tên chỉ định cho trình kết nối

Để register connnect cần gửI data config vào http://localhost:8083/connectors/



Cấu hình Consumer để nhận dữ liệu từ kafka



**VD chạy các câu lệnh thử nghiệm vớI replication =3 và stop 1 container kafka xem có mất dữ liệu trước đó không khi 1 group khác read topic**

**1: docker-compose up -d : run các container trong file docker**

**2: exec vớI postgres container**

**- Truy cập vào vùng chứa và thực hiện input terminal**

docker exec -it <id\_container> bash

**- Cấu hình host, post, user name**

psql -h localhost -p 5432 -U postgres

**- tạo schema và db, thực hiện insert tạo các mesage**

CREATE SCHEMA debezium;

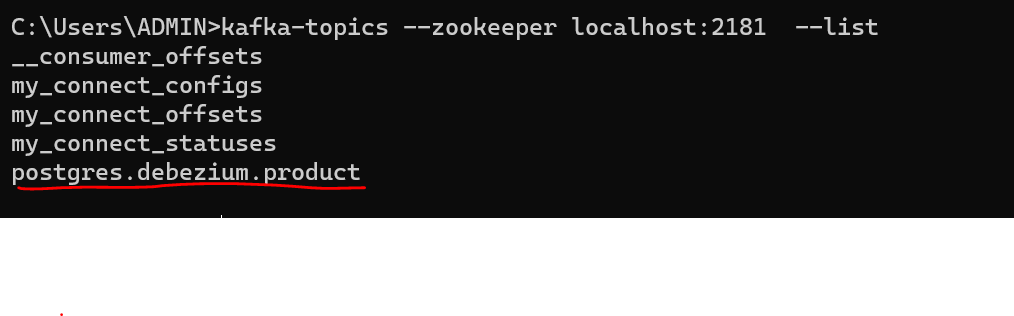
CREATE TABLE debezium.product (pro int);

insert into debezium.product values (23);

**3: đăng ký connect sẽ tạo ra topic**

**//danh sach topic**

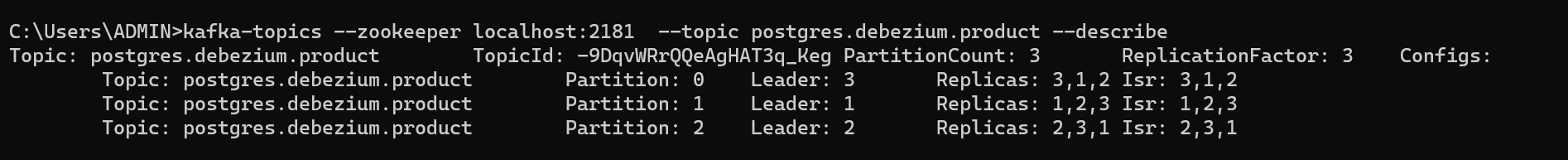
kafka-topics --zookeeper localhost:2181 --list



Xem chi tiết topic:

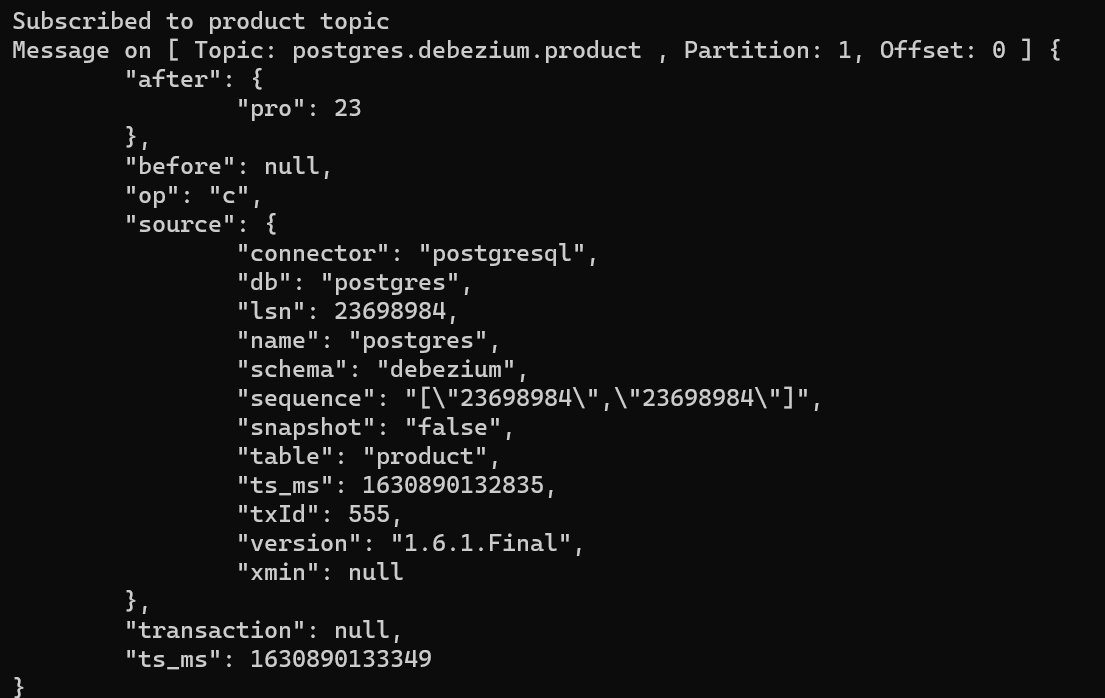
**//thong tin (describe) ve topic**

kafka-topics --zookeeper localhost:2181 --topic <name> --describe



**Thông tin bao gồm các partition đc replicate sang các broker id và đc đồng bộ (Isr) ở các broker id và leader là các id của Broker chứa partition đó**

4:thông tin mesage đc đọc ra



**5: Partition đang là chính ở broker id = 1, nên stop kafka-1 rồI dùng group đọc lạI mesage**

**Xem còn dữ liệu không**

**zookeeper Tự động quản lý chuyển leader từ brocker đc đồng bộ (Isr) vẫn còn partition nên khi đọc lạI vẫn còn**

