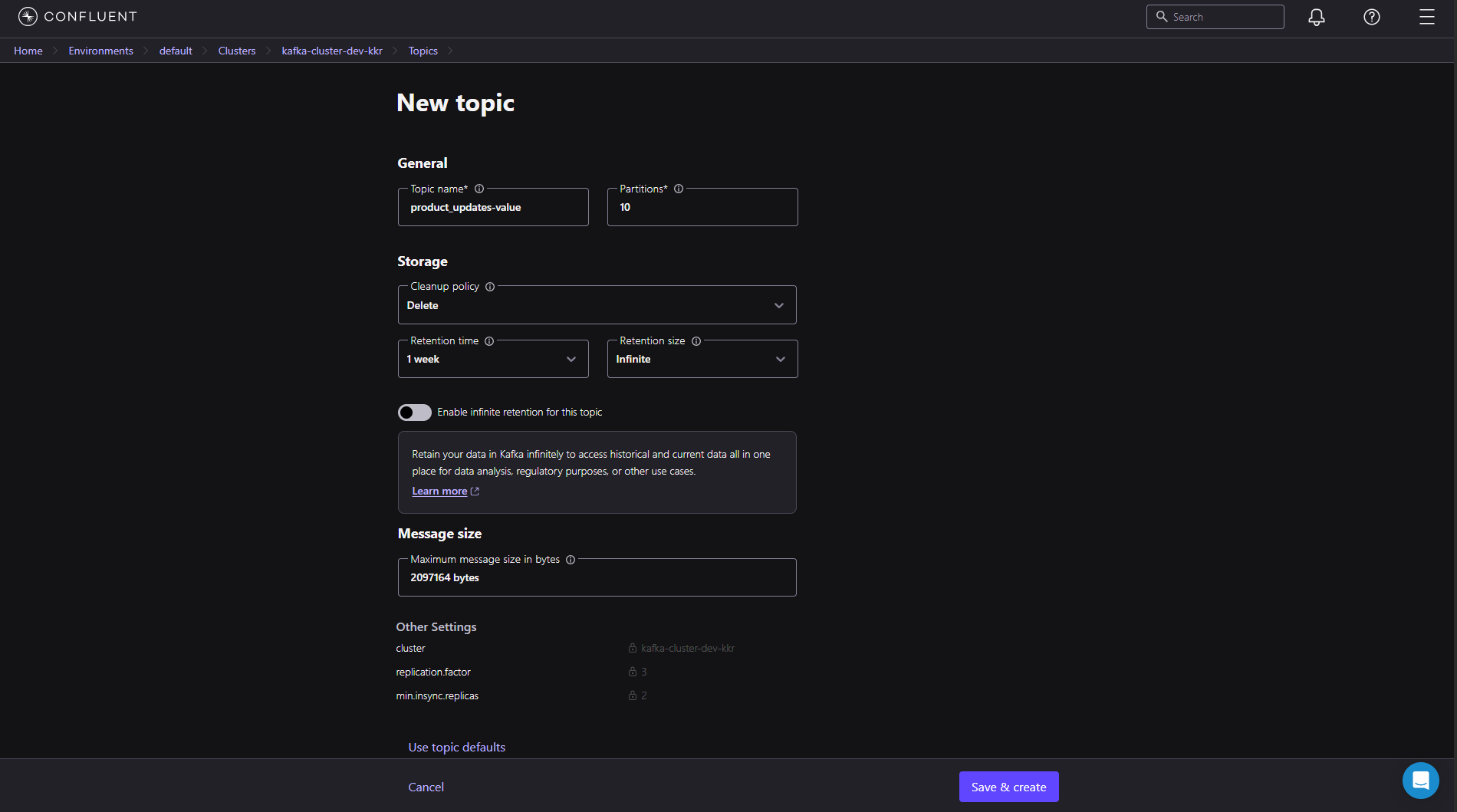
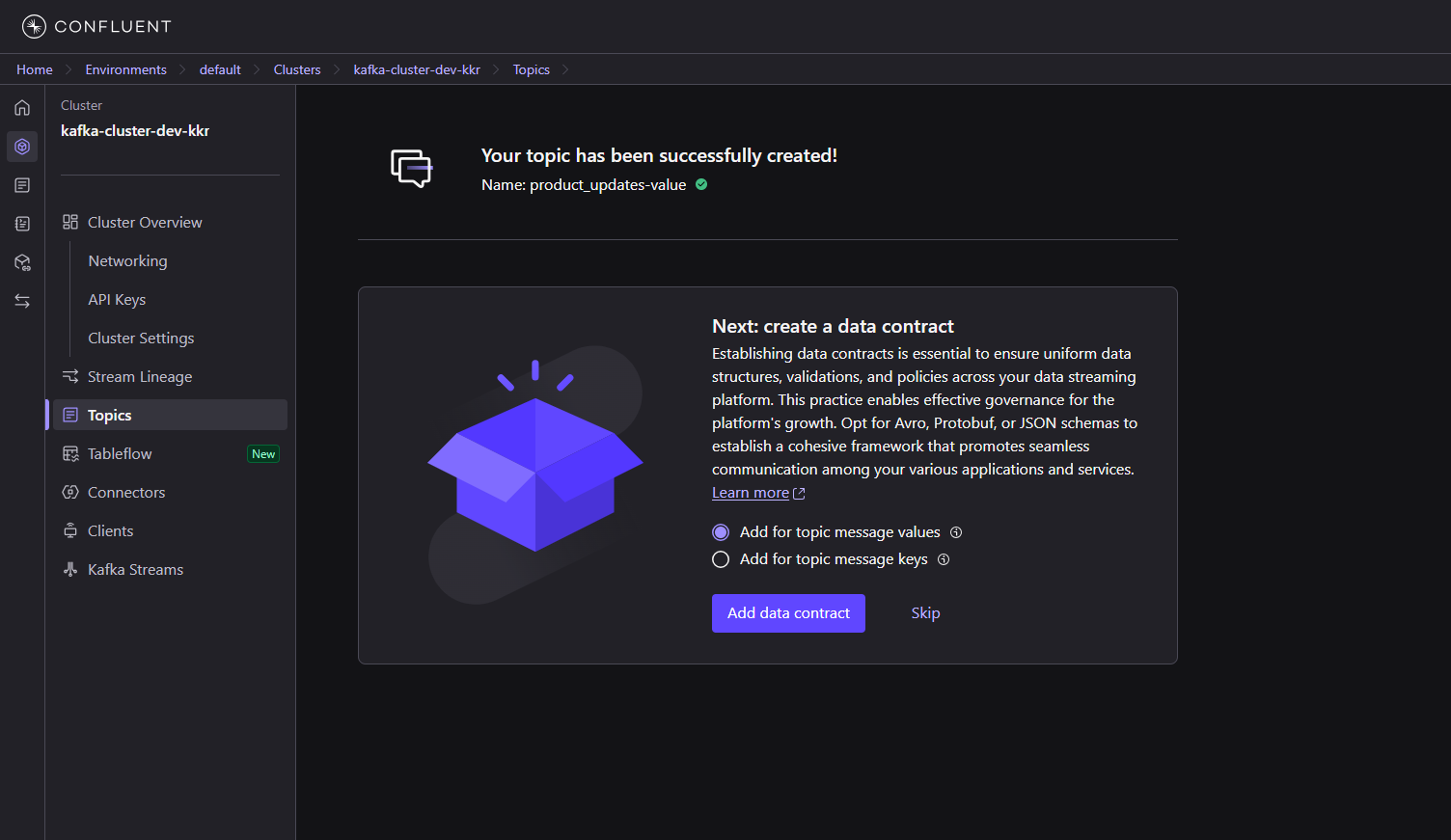
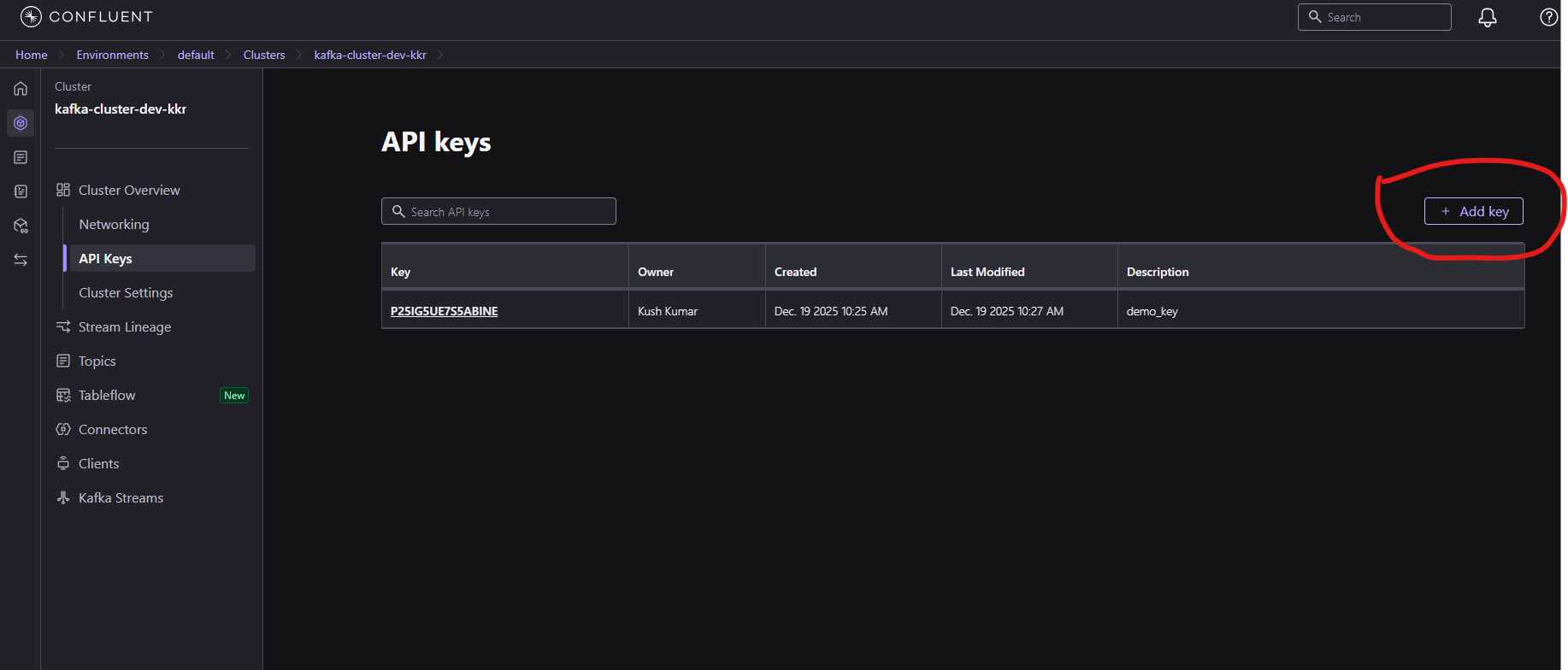
KAFKA ASSIGNMENT DOCUMENTATION:

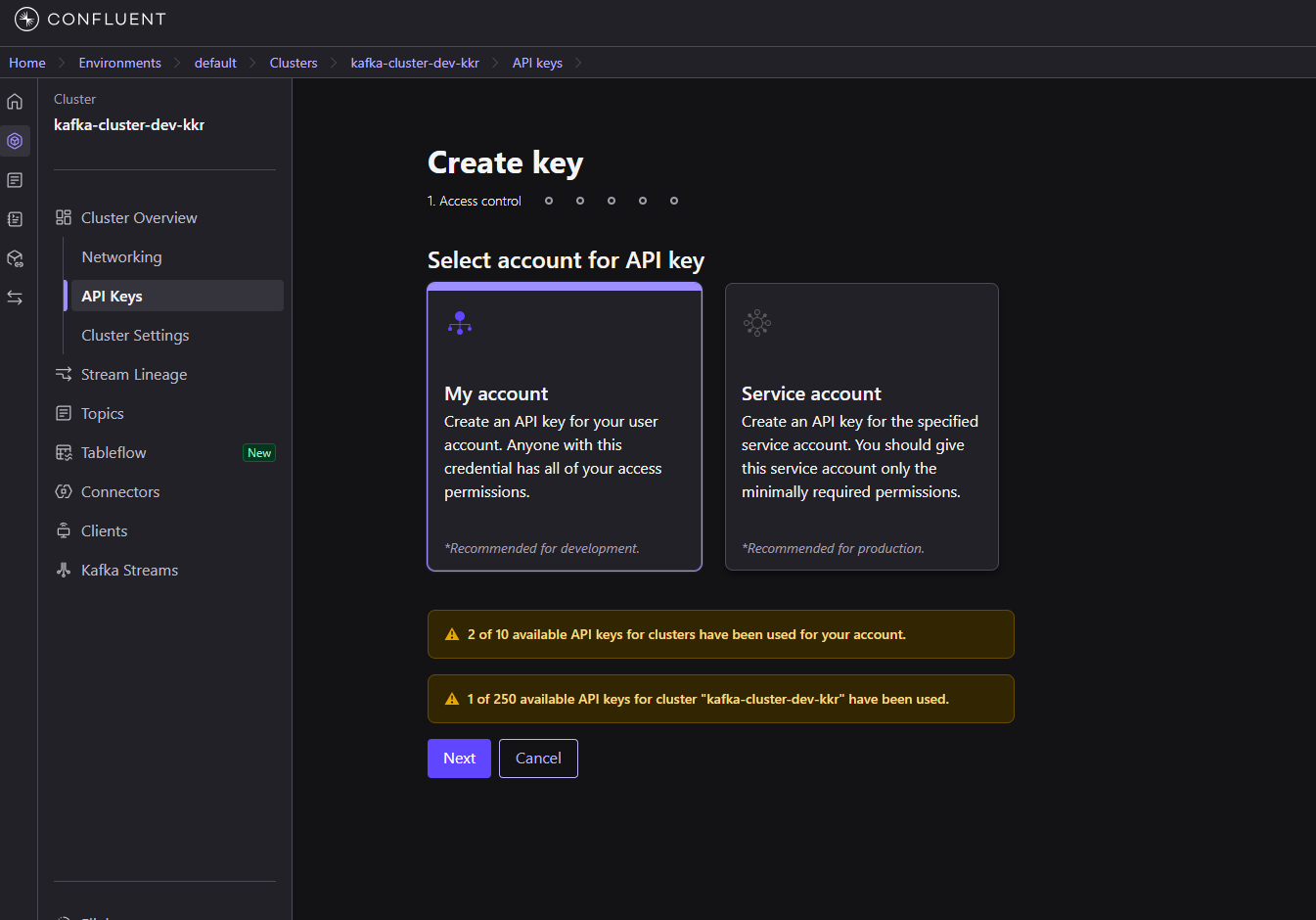
Create a topic : product\_updates-value with 10 partitions in cluster: kafka-cluster-dev-kkr

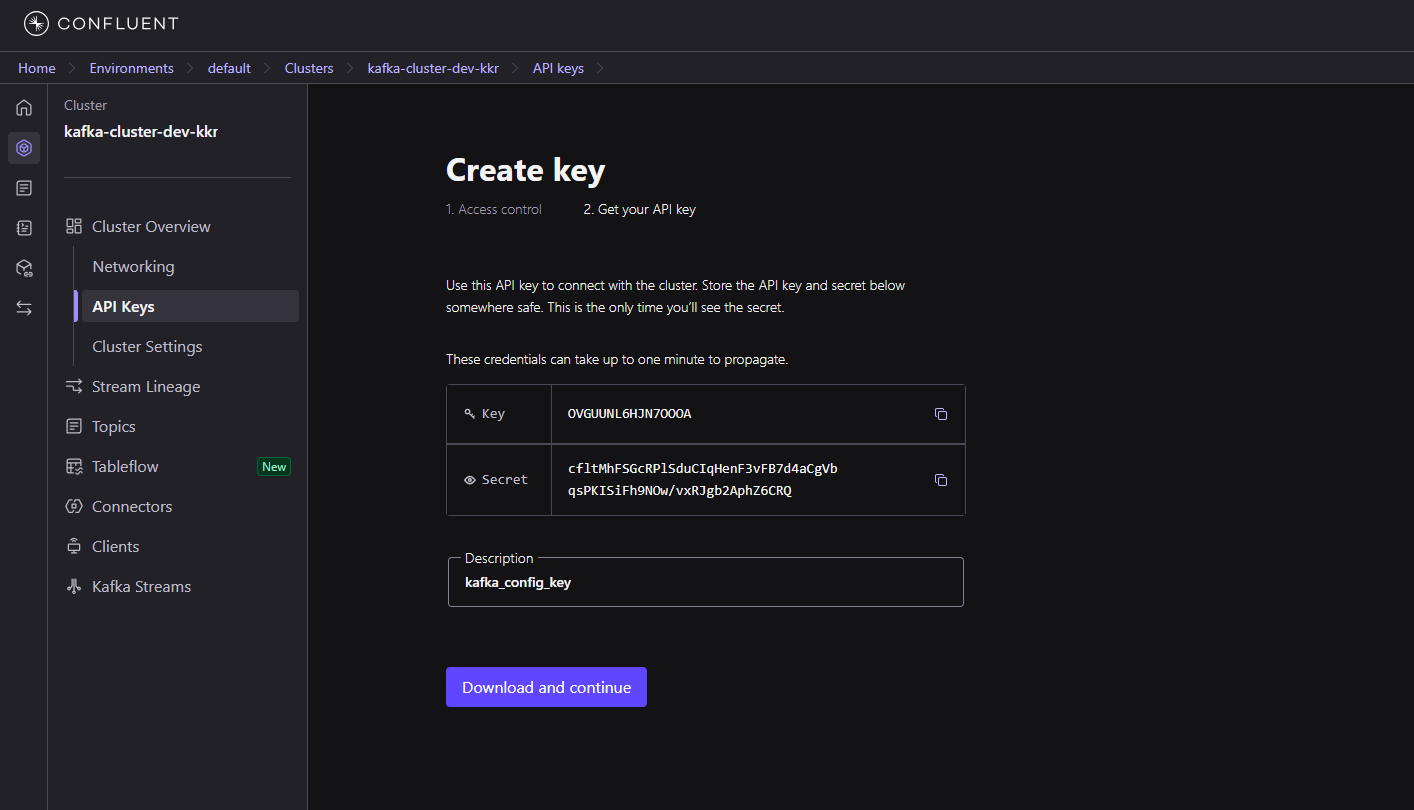


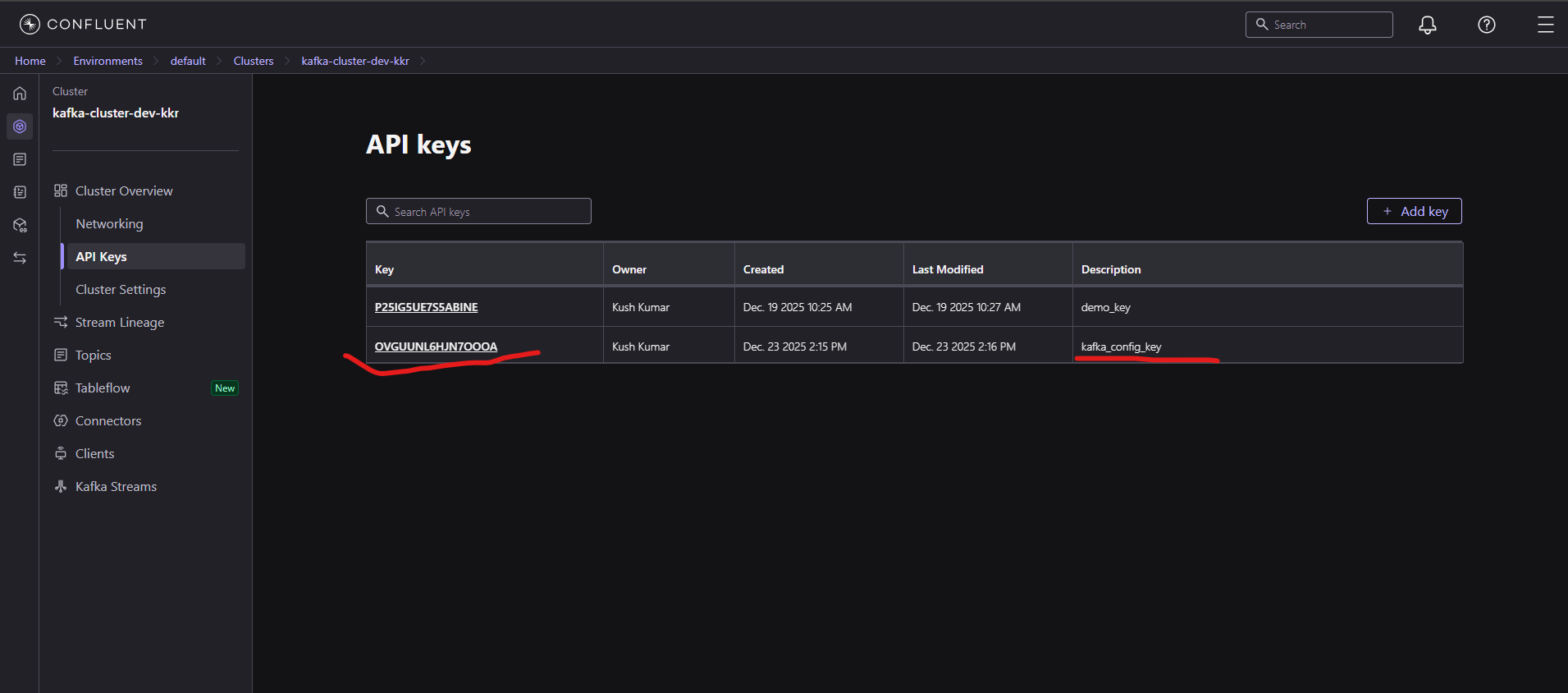


Generate API for kafka config :

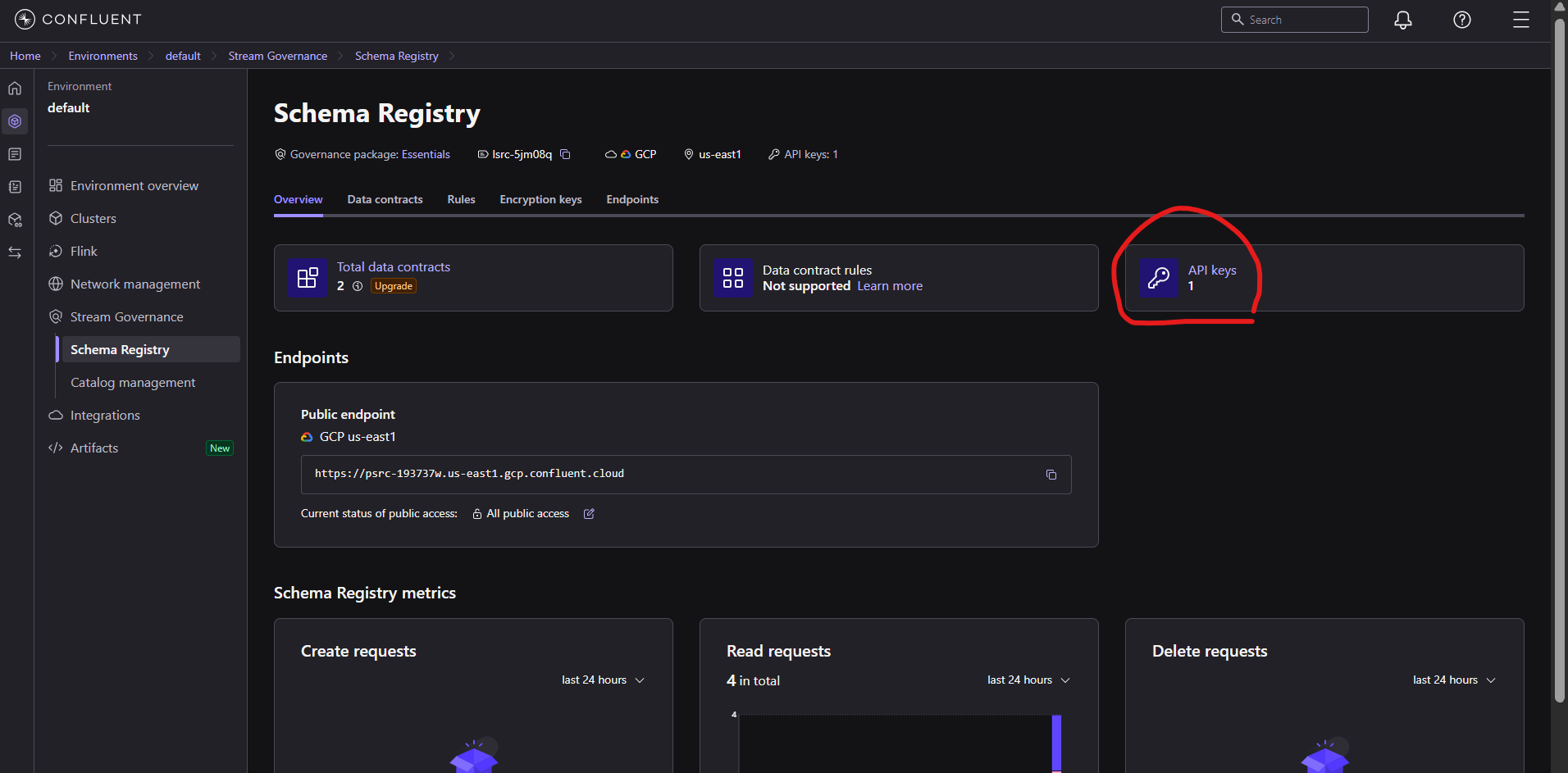


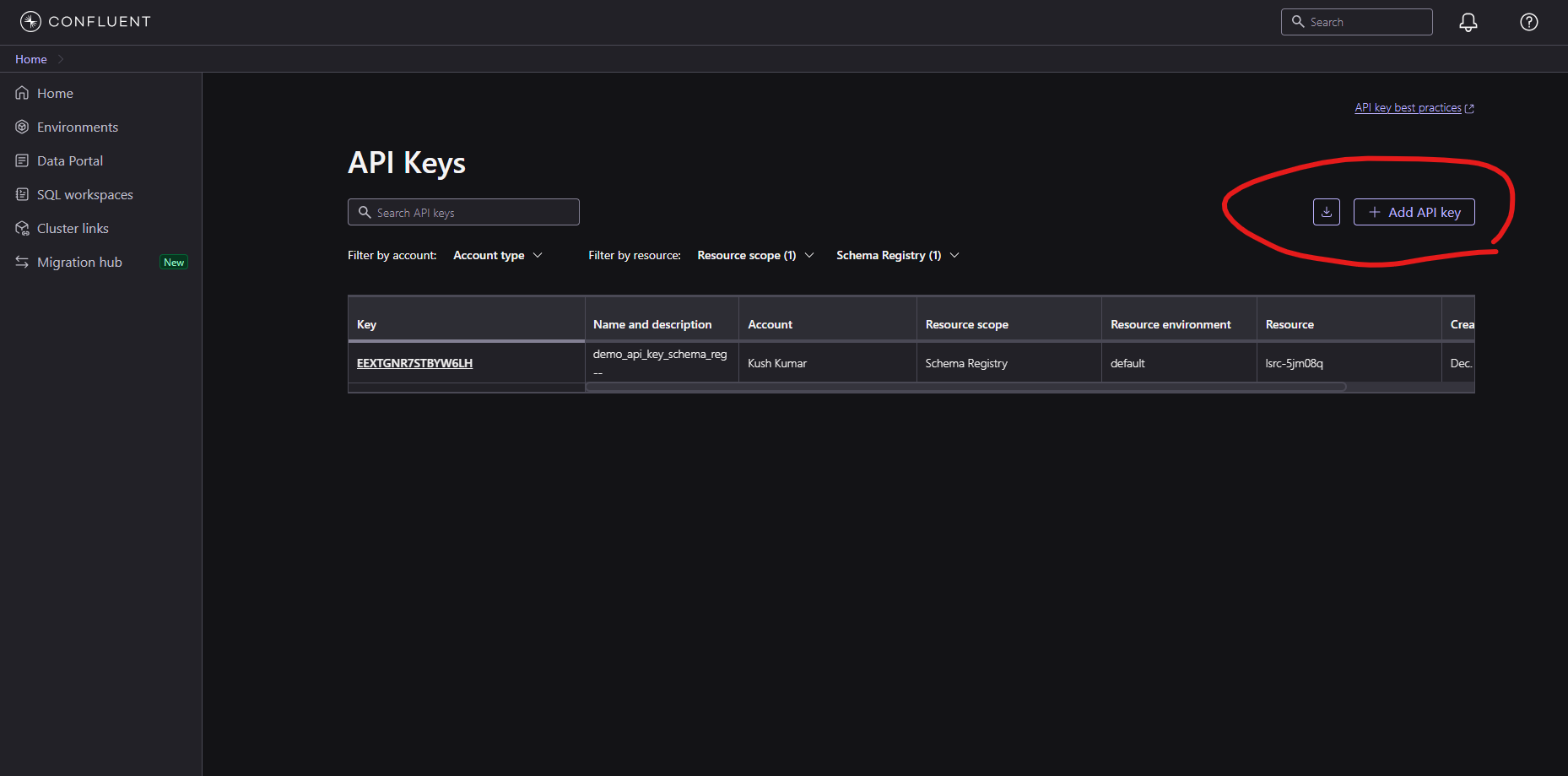


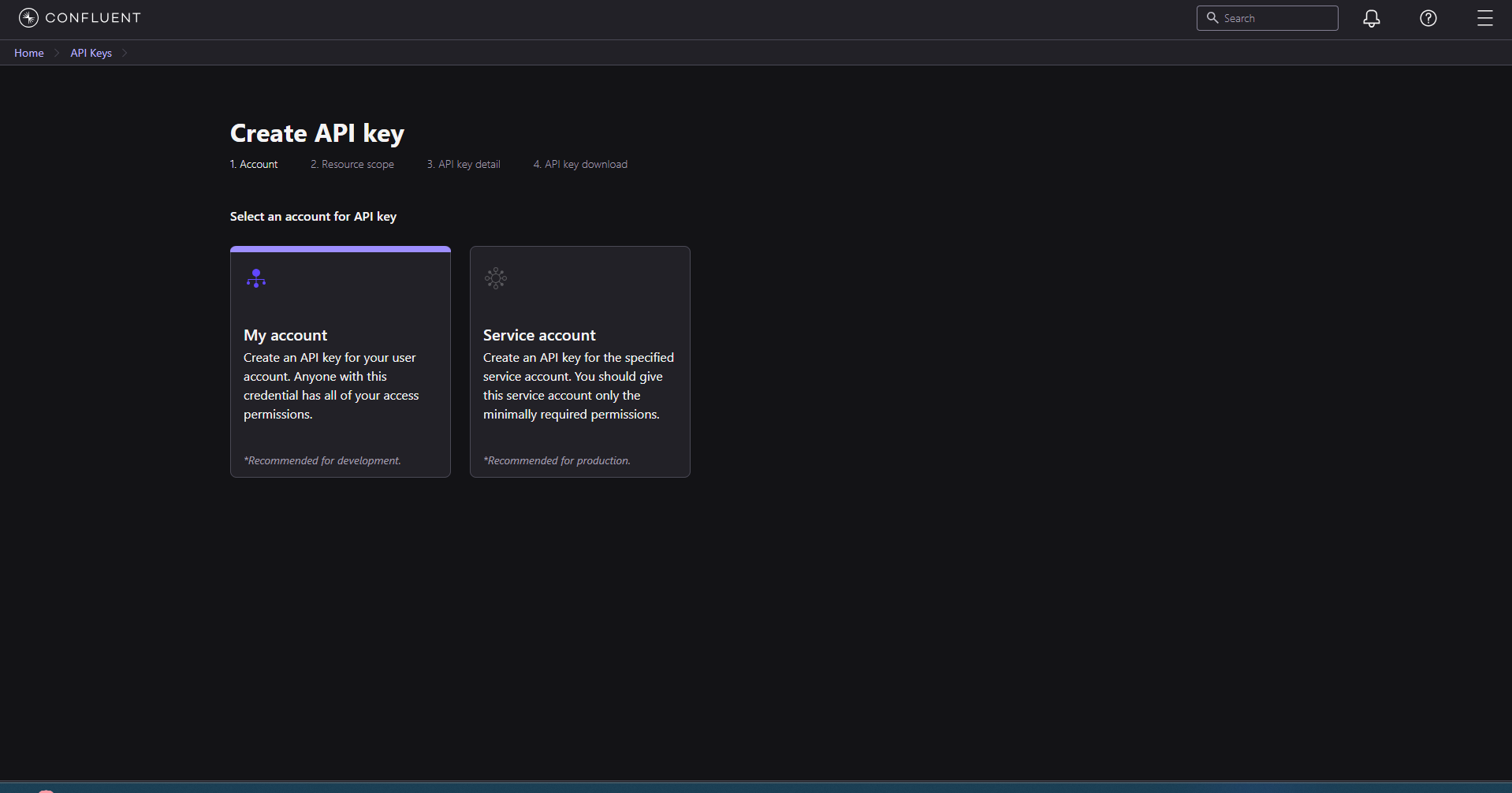


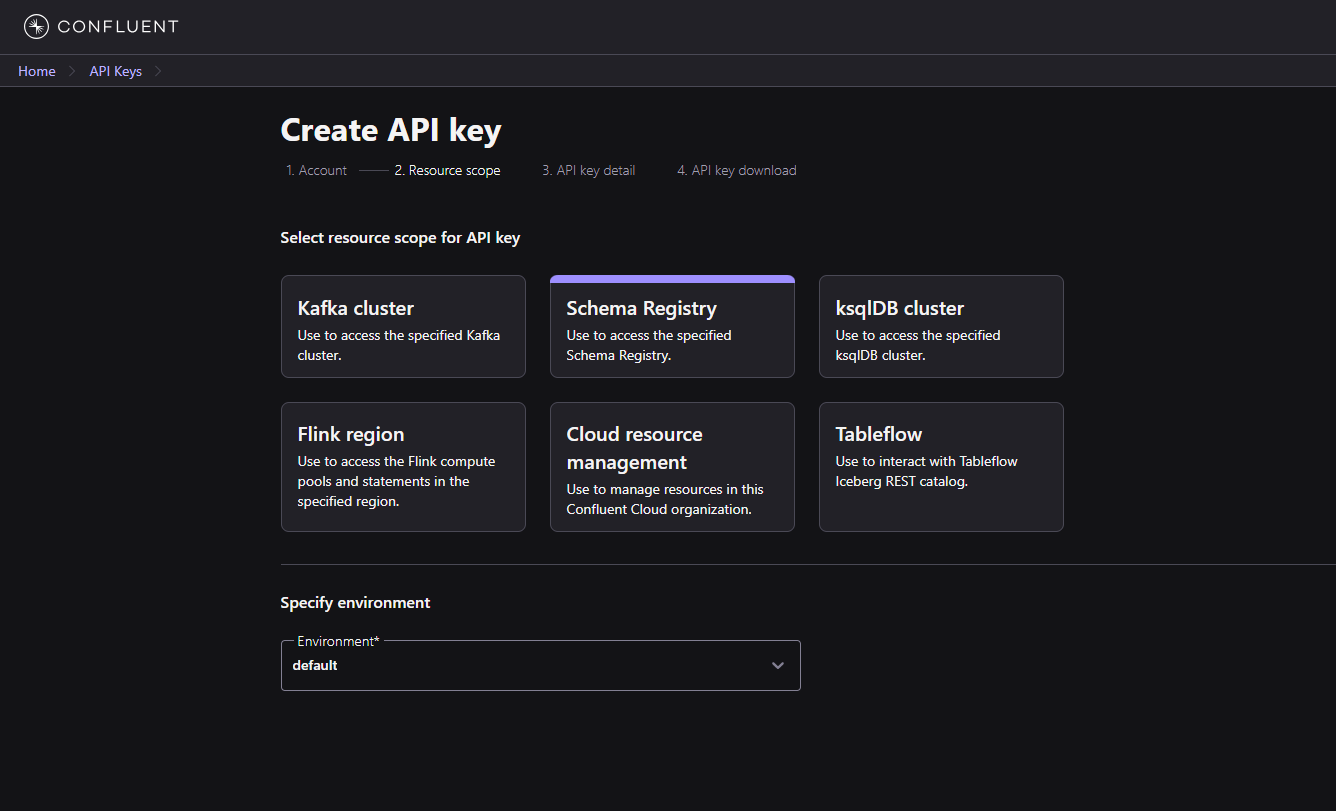


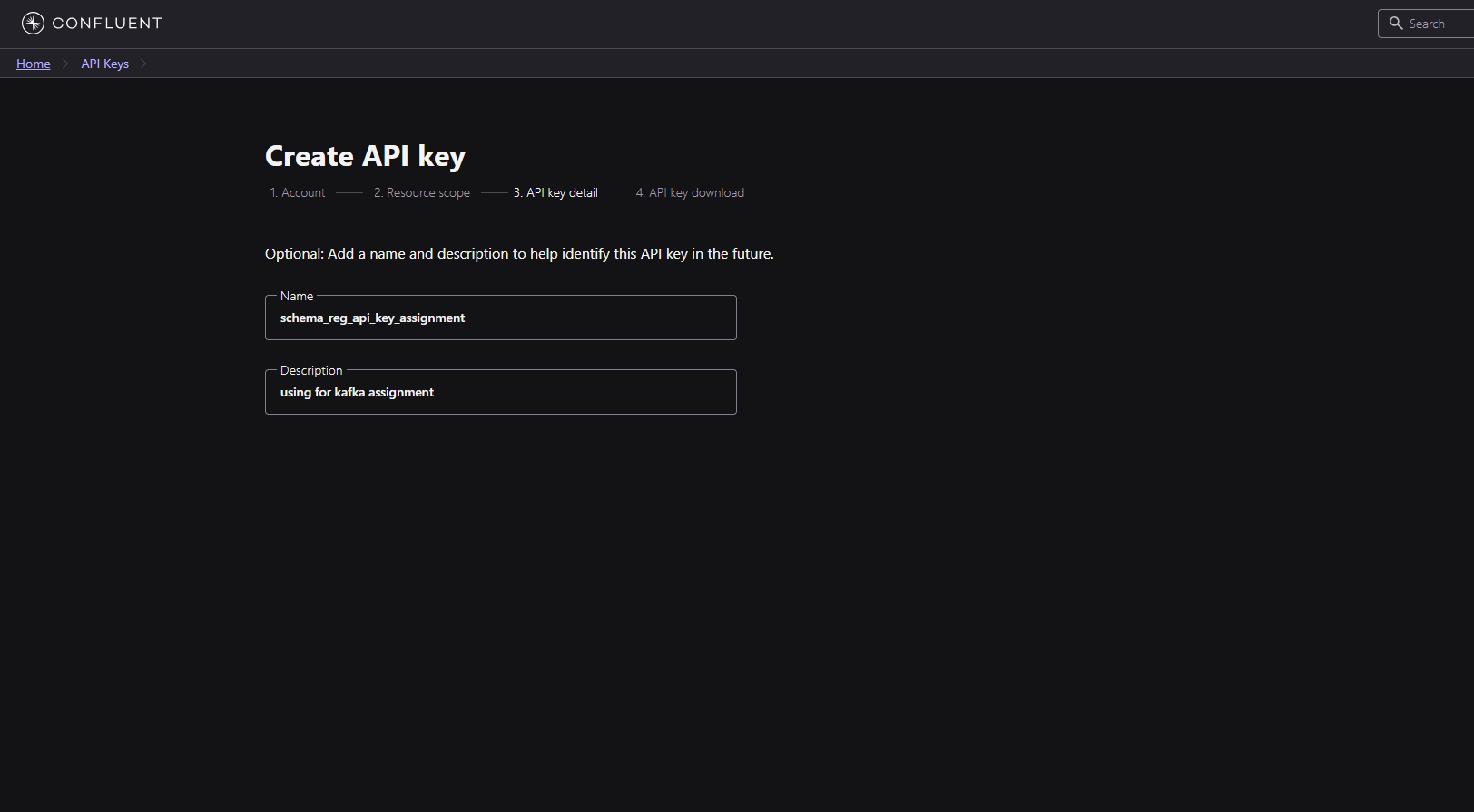
Generate API for schema registry :

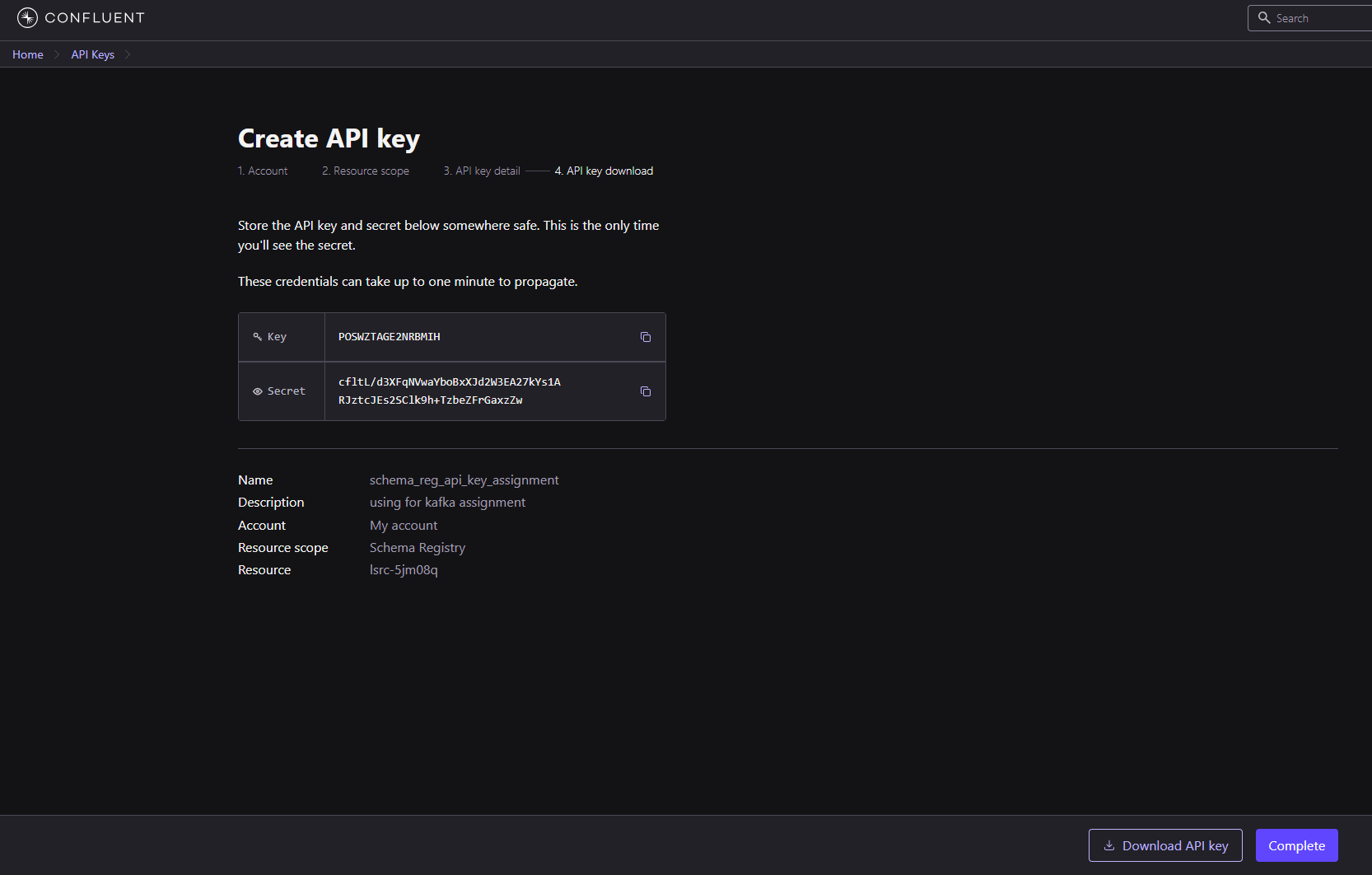


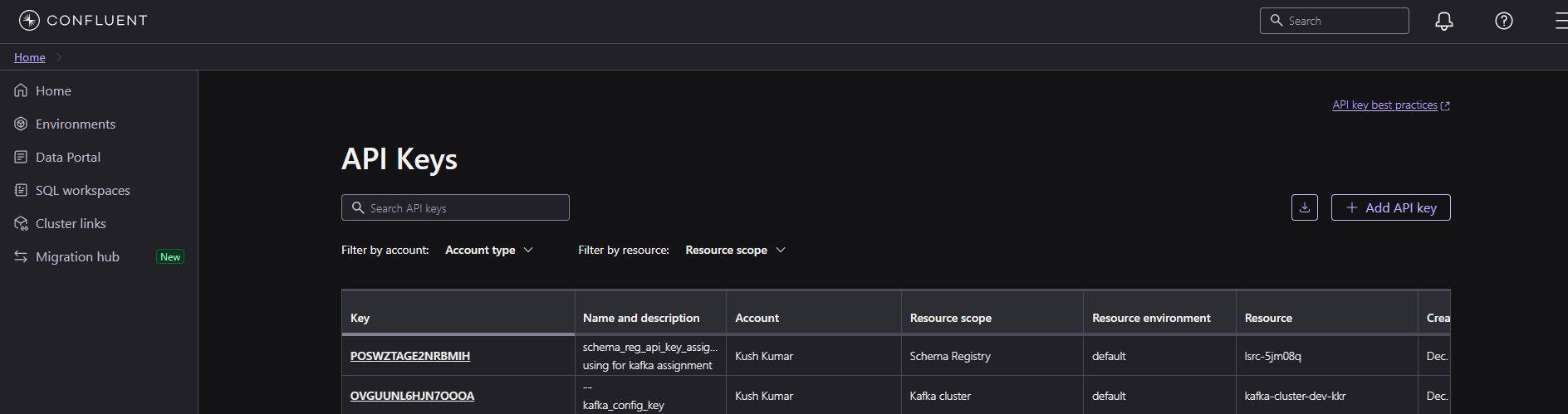






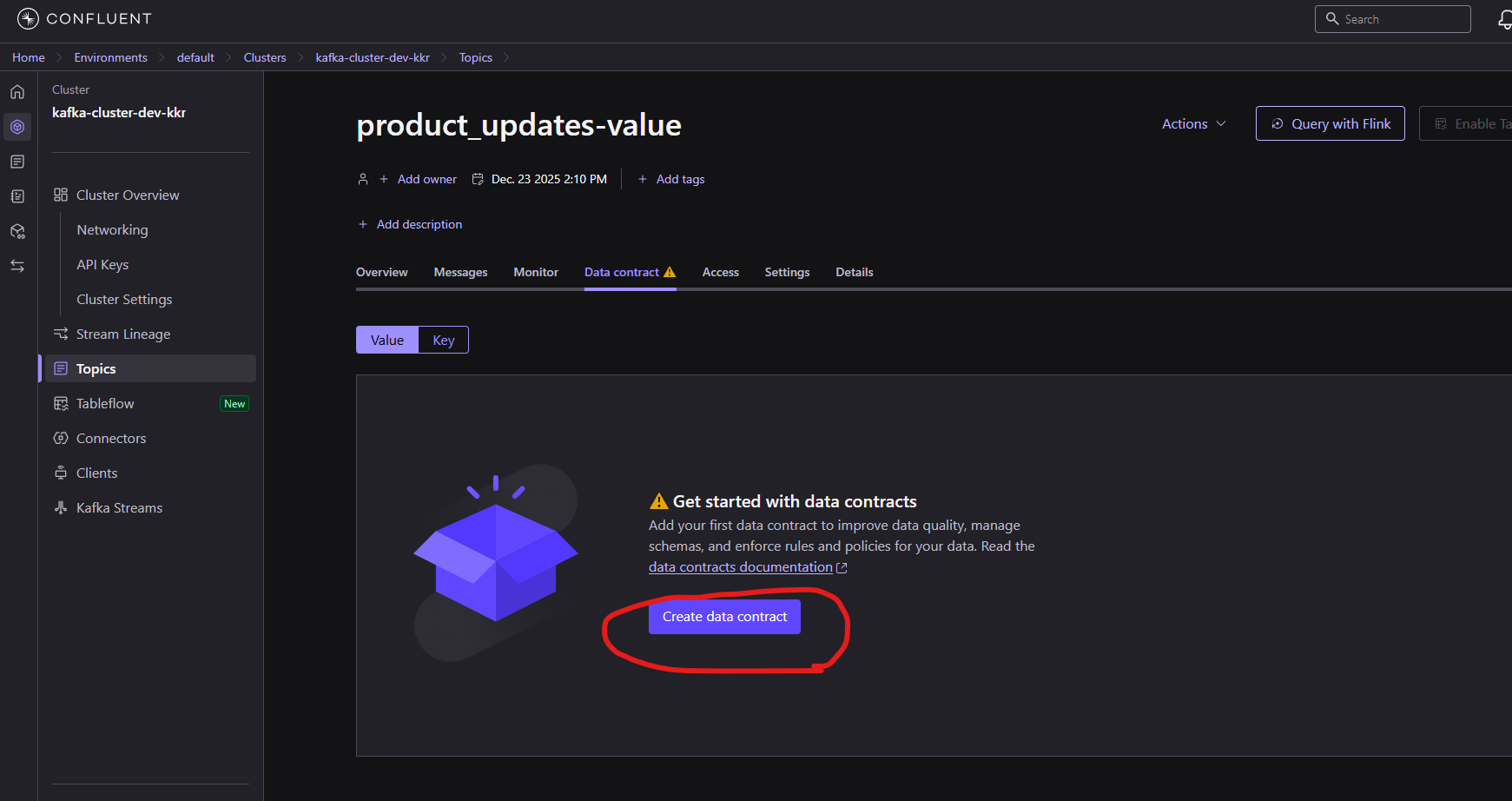


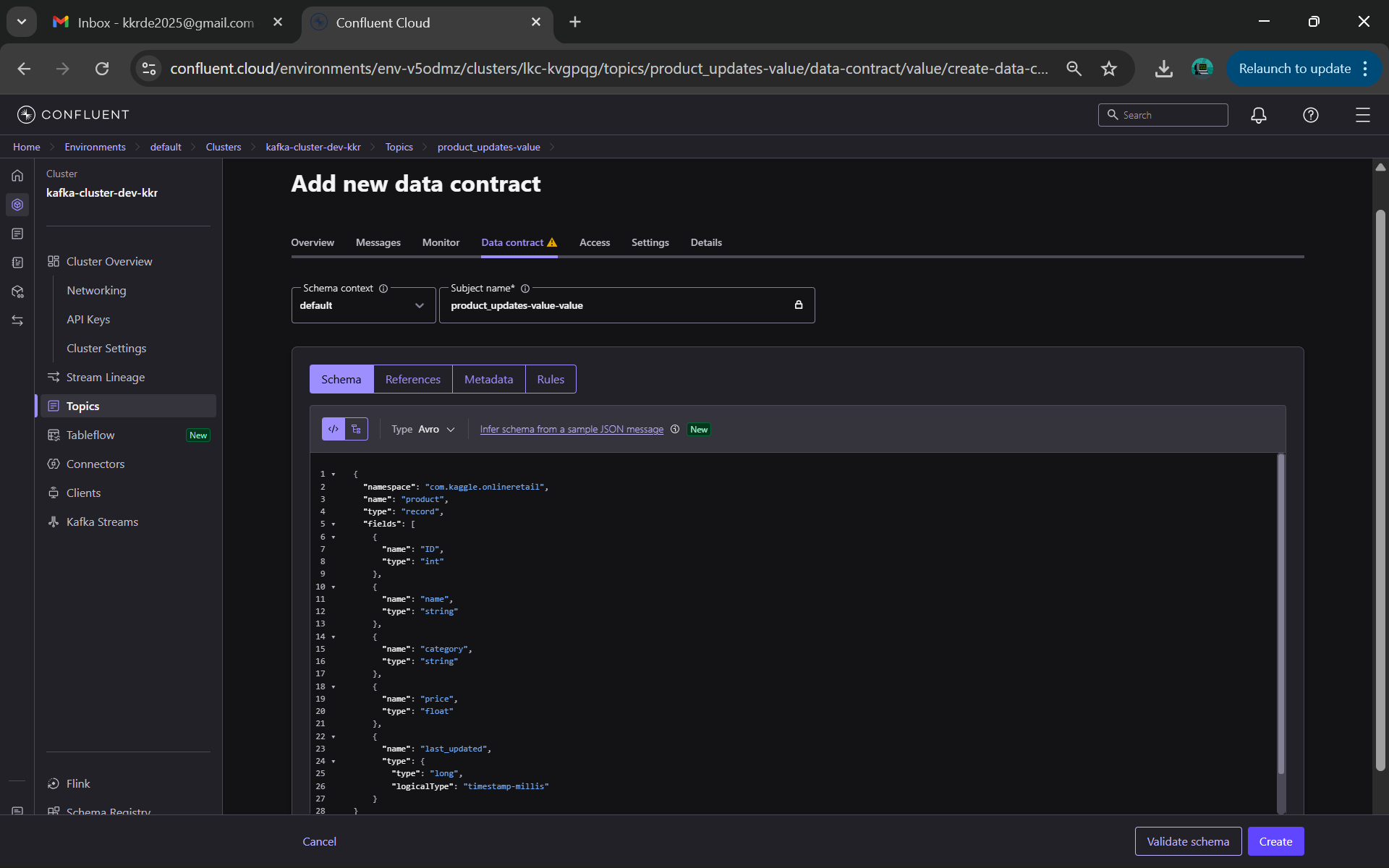


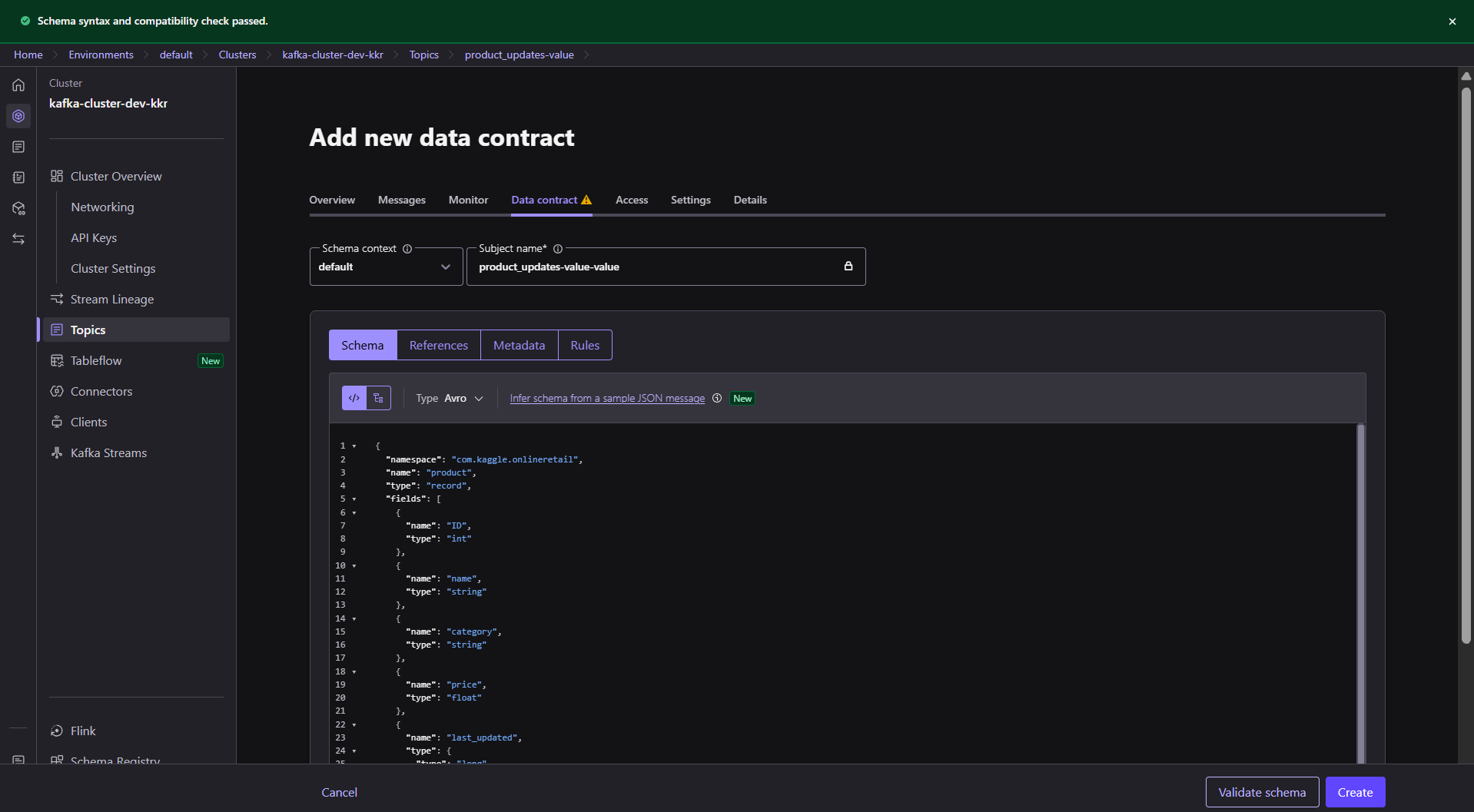


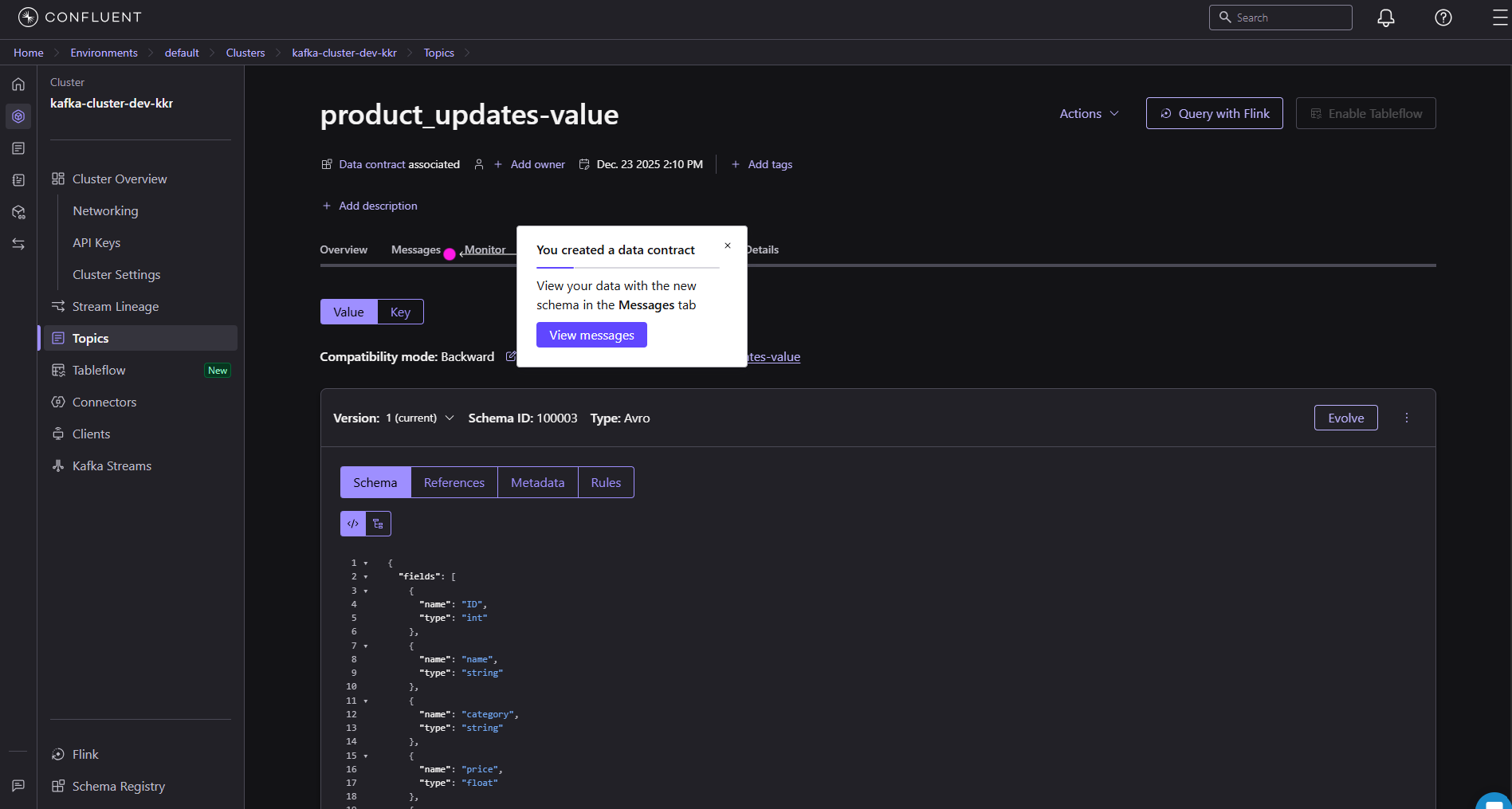
Copy and paste details from both generated keys to both producer and consumer code.

Copy and paste the AVRO schema in data contract:

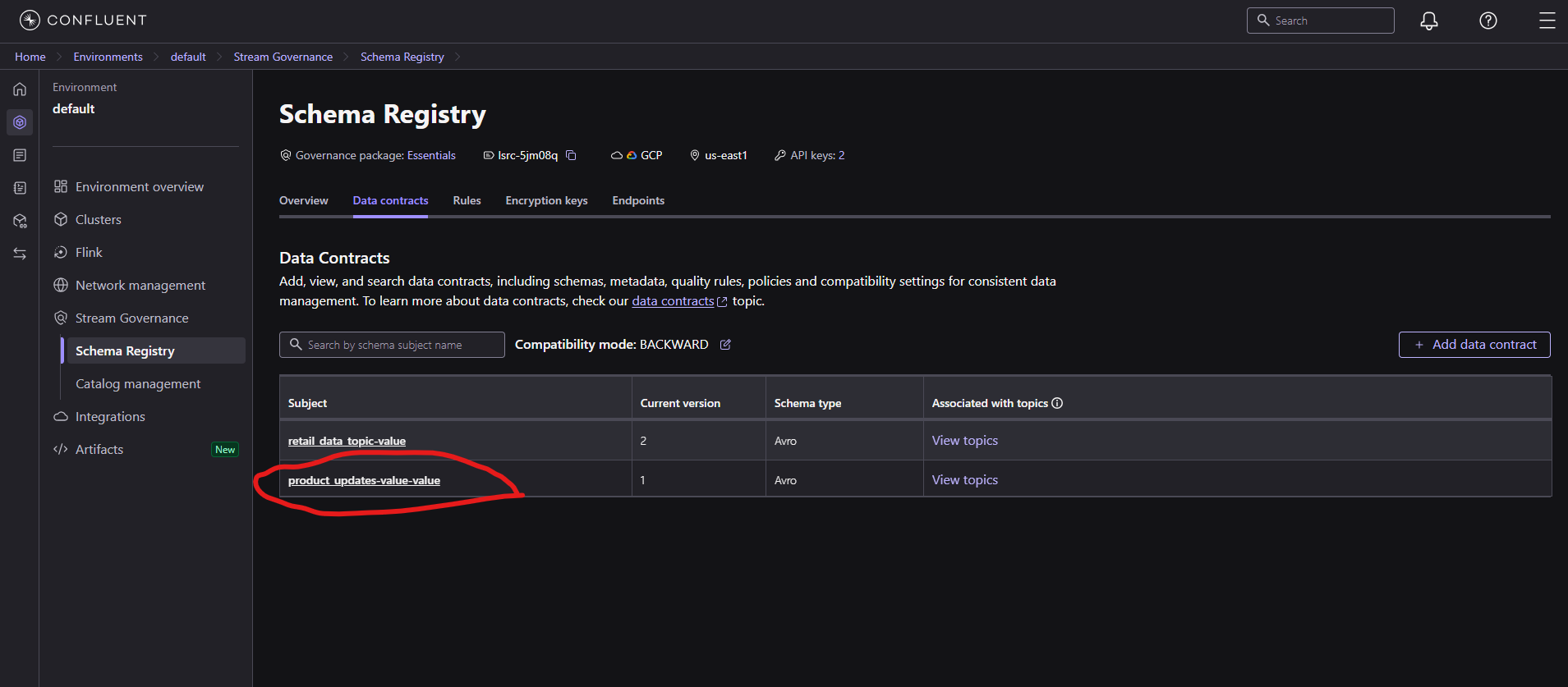


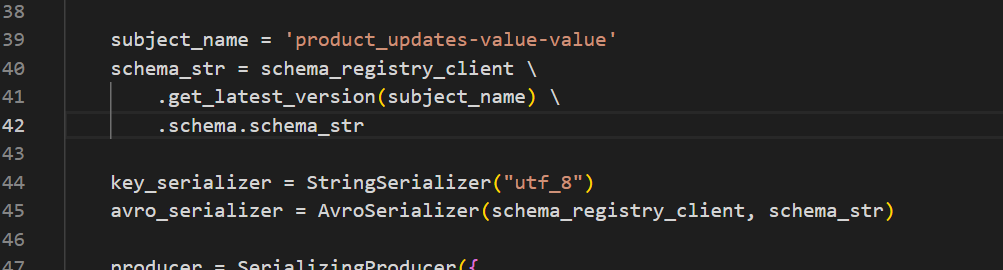






Copy the subject name (by going to>schema registry>data contract) to subject name in producer code:





Create table products in kafka\_assignment database:

CREATE TABLE products (

ID INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

category VARCHAR(255),

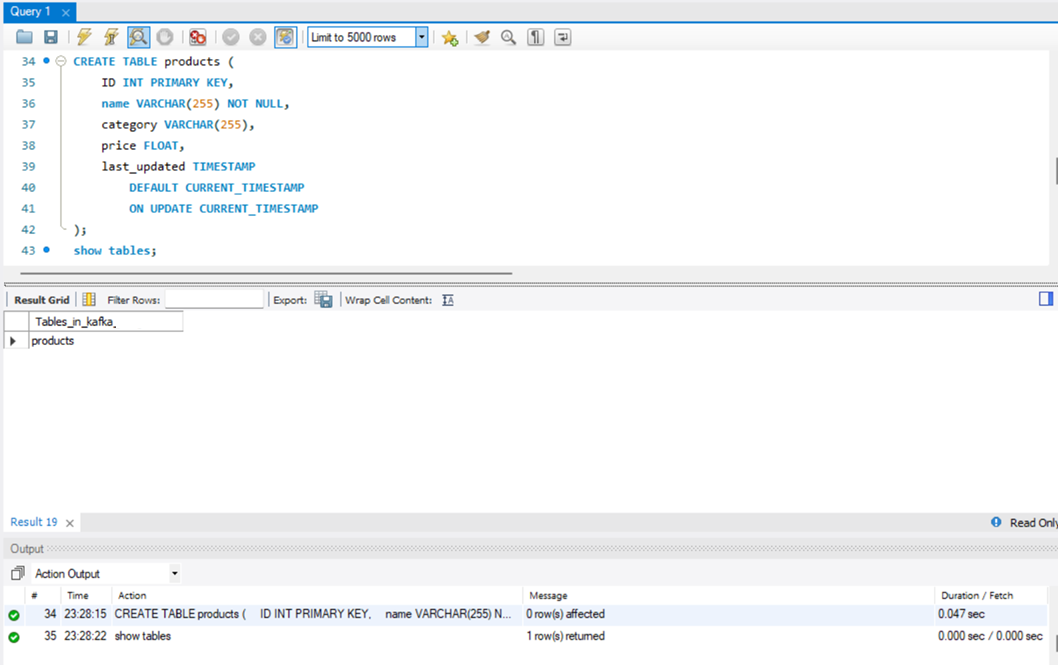
price FLOAT,

last\_updated TIMESTAMP

DEFAULT CURRENT\_TIMESTAMP

ON UPDATE CURRENT\_TIMESTAMP

);



step 1. Load some rows into the products table.

INSERT INTO Products (ID, name, category, price, last\_updated) VALUES

(1, 'Laptop', 'Electronics', 75000.50, '2025-01-10 10:15:00'),

(2, 'Smartphone', 'Electronics', 45000.00, '2025-01-11 11:20:00'),

(3, 'Headphones', 'Accessories', 2500.75, '2025-01-12 09:30:00'),

(4, 'Keyboard', 'Accessories', 1500.00, '2025-01-12 14:45:00'),

(5, 'Mouse', 'Accessories', 800.50, '2025-01-13 16:10:00'),

(6, 'Office Chair', 'Furniture', 12000.00, '2025-01-14 13:05:00'),

(7, 'Desk', 'Furniture', 18000.25, '2025-01-14 17:40:00'),

(8, 'Monitor', 'Electronics', 22000.00, '2025-01-15 10:55:00'),

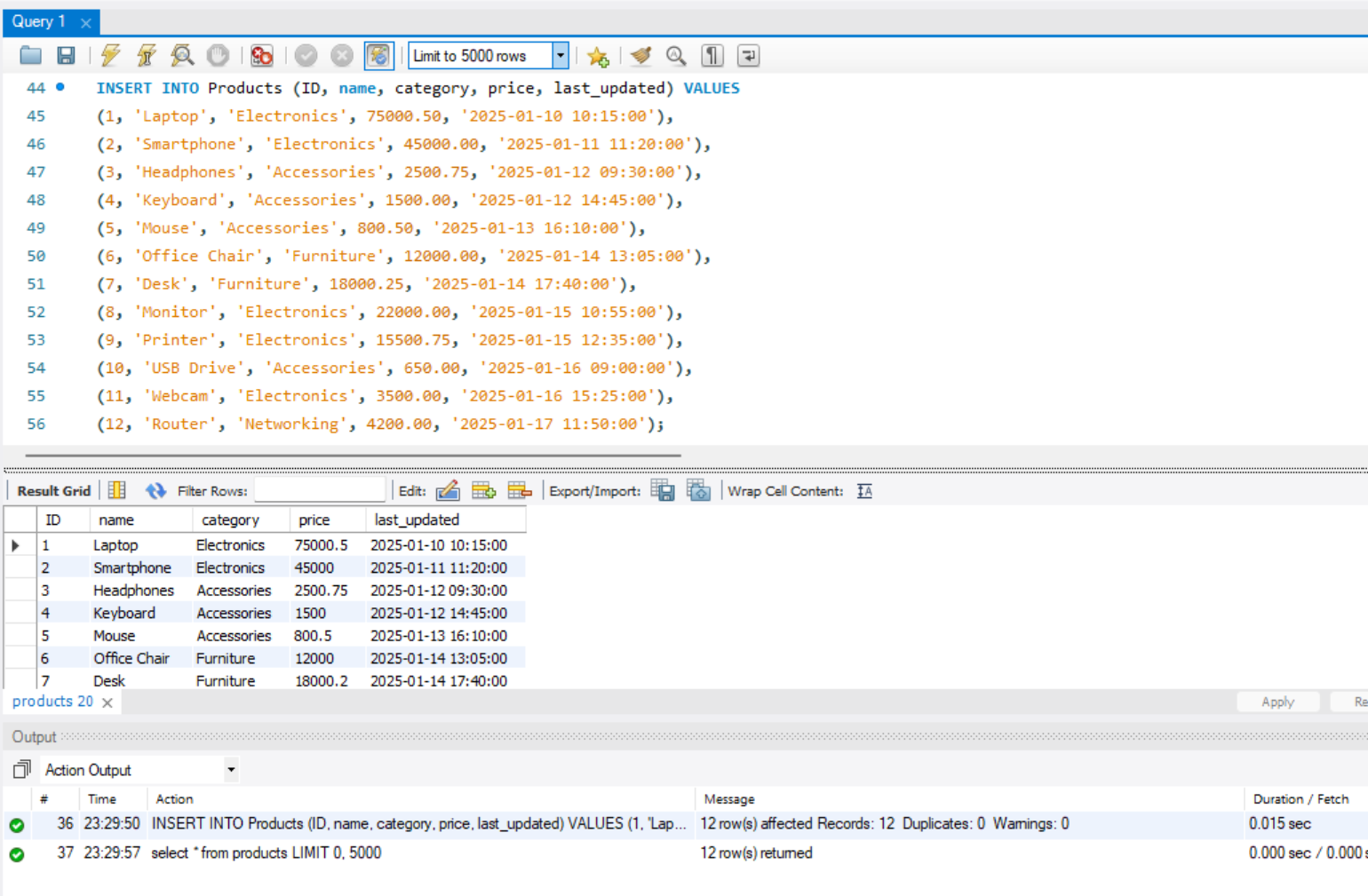
(9, 'Printer', 'Electronics', 15500.75, '2025-01-15 12:35:00'),

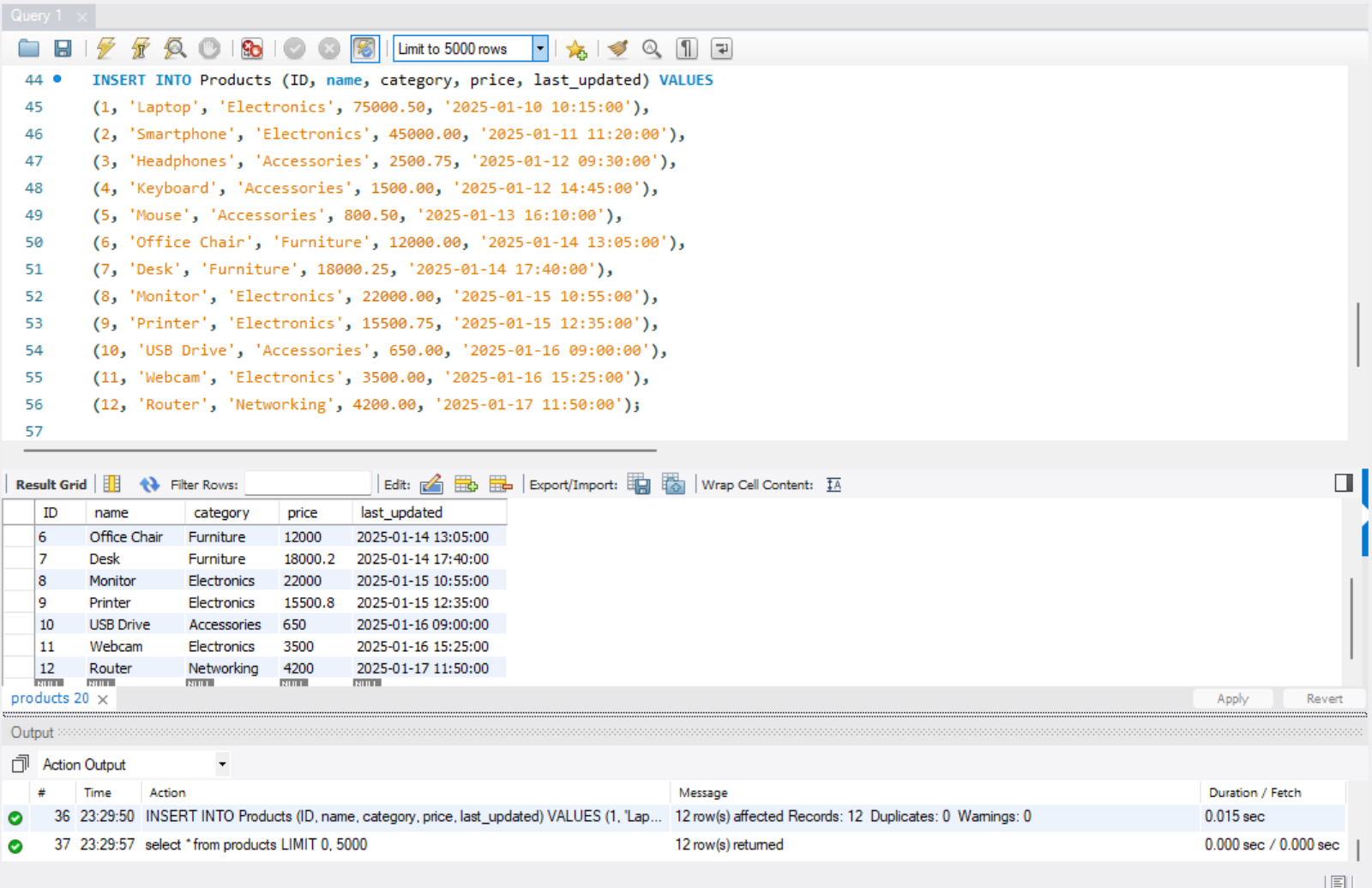
(10, 'USB Drive', 'Accessories', 650.00, '2025-01-16 09:00:00'),

(11, 'Webcam', 'Electronics', 3500.00, '2025-01-16 15:25:00'),

(12, 'Router', 'Networking', 4200.00, '2025-01-17 11:50:00');

Mysql screenshots :

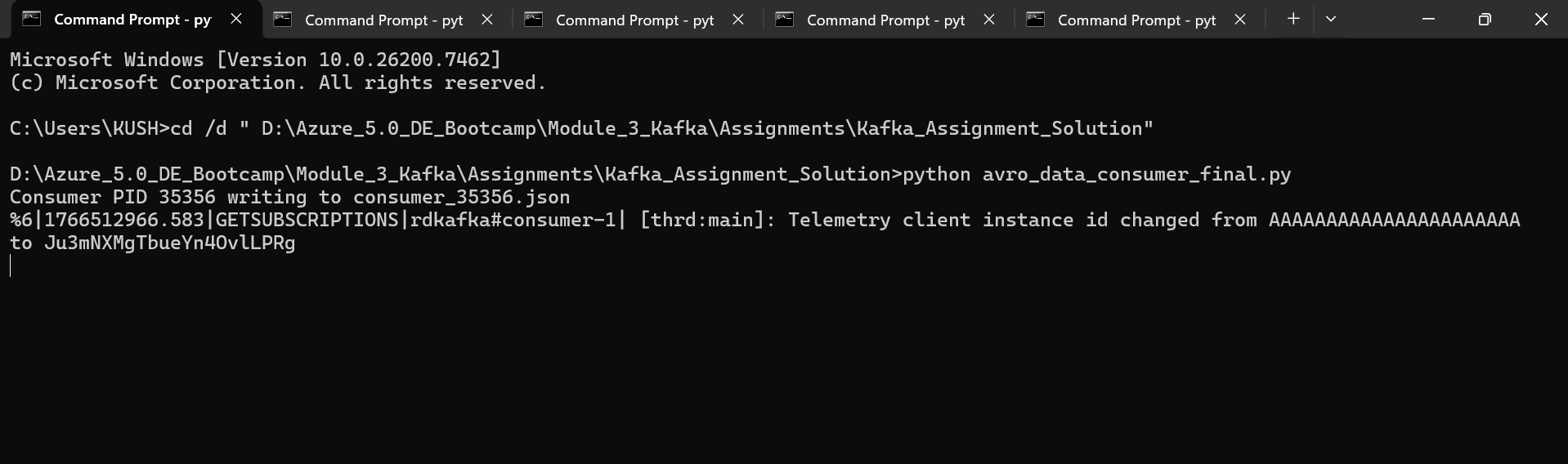


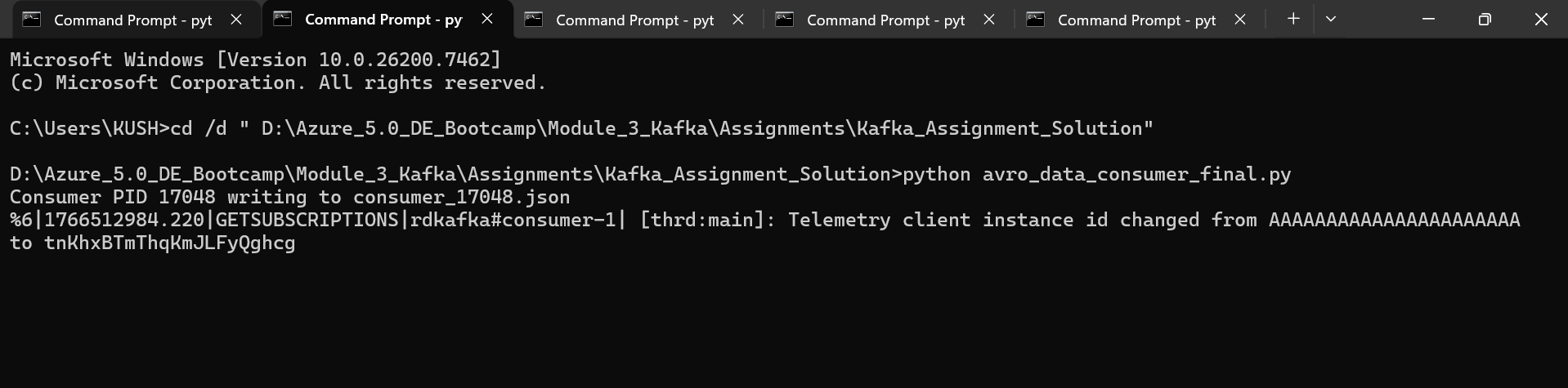


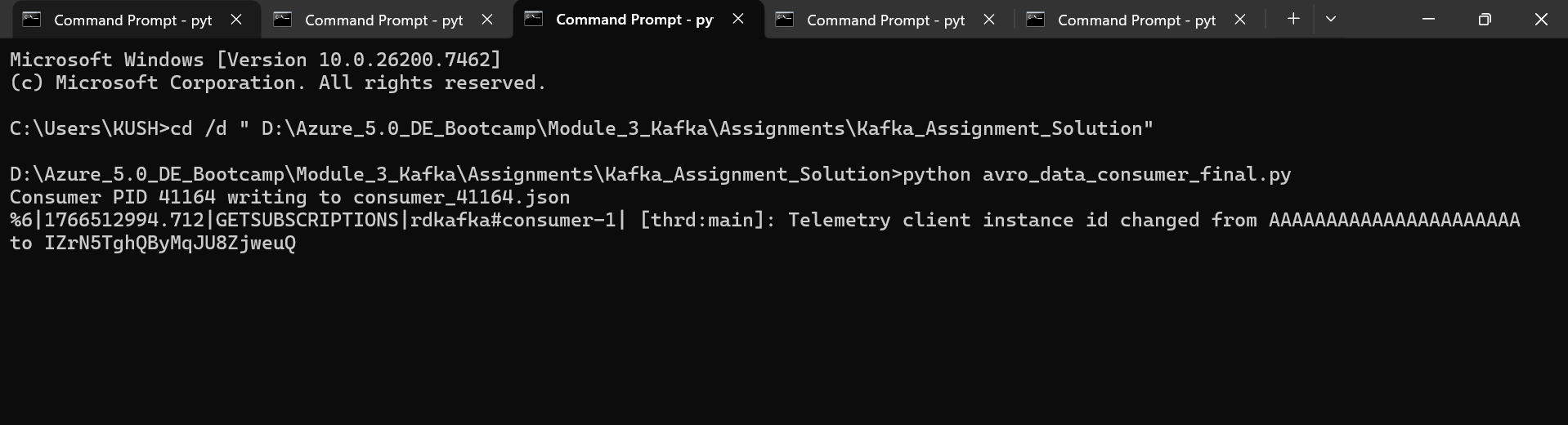
Initially loaded 12 records in the table ‘Products’ in MySQL database.

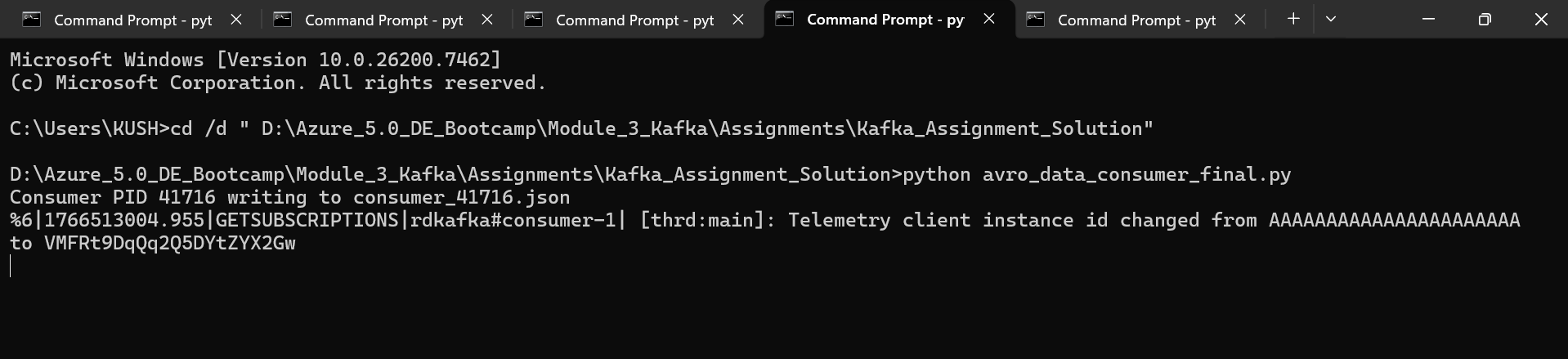
step 2. Start 5 instances of consumer code in parallel with default offset mode.

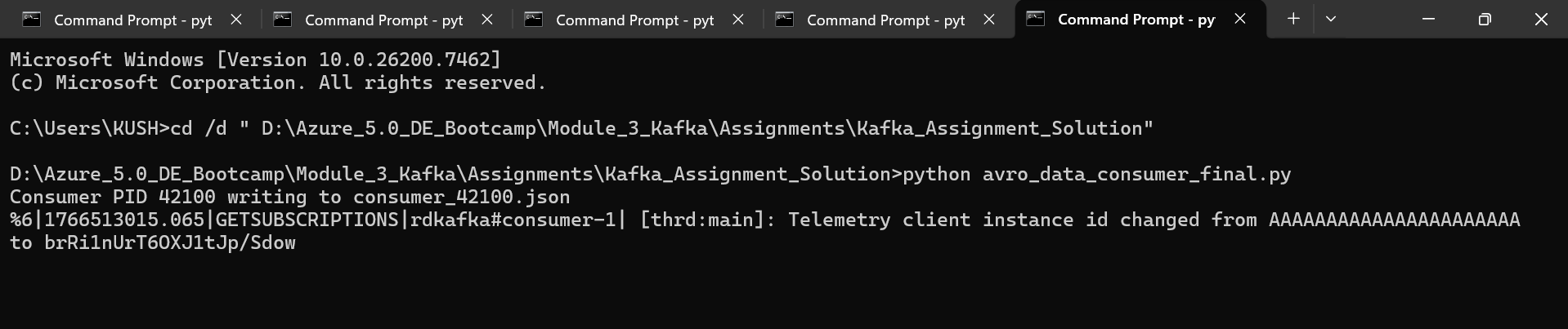
Consumer screenshots :





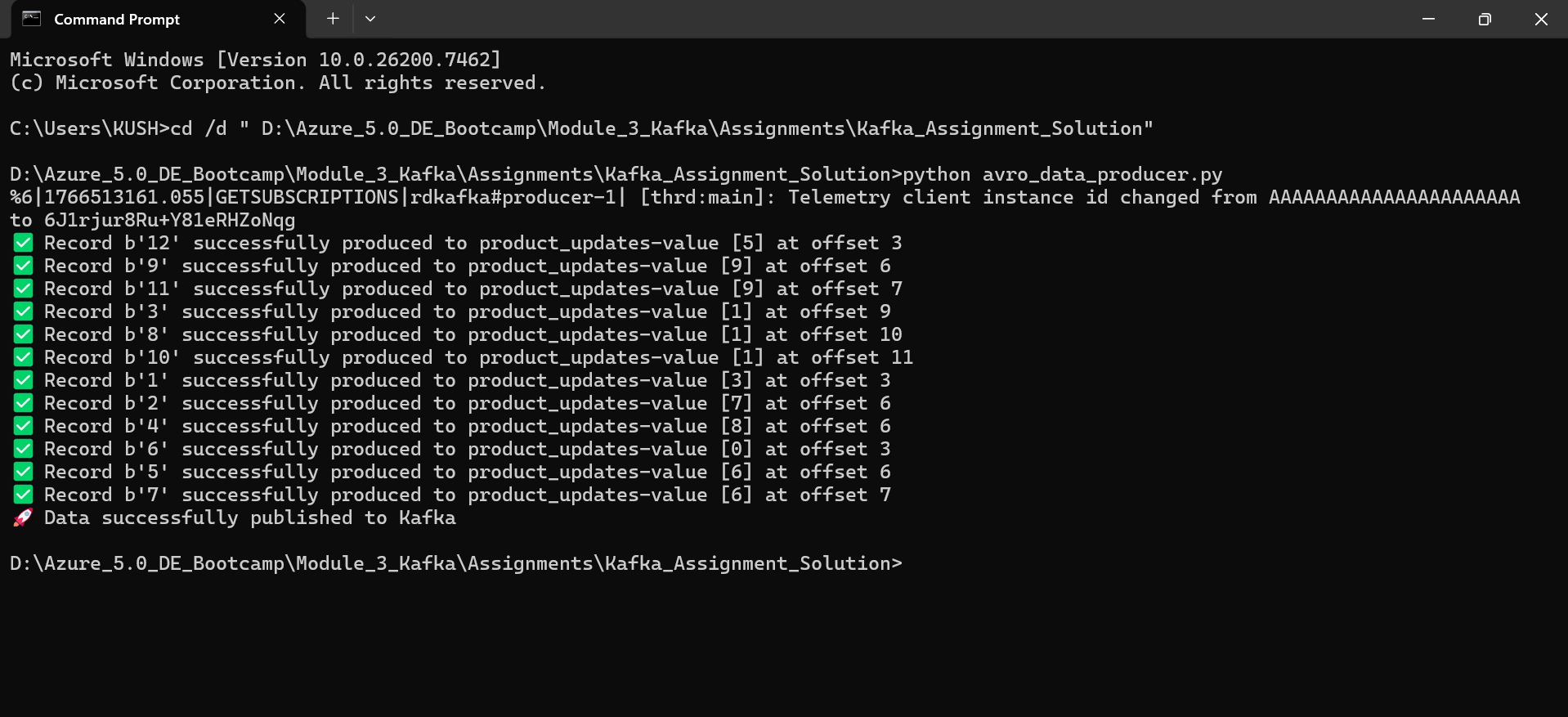






step 3. Run the producer code to load the data from Mysql table into producer then it will publish to kafka topic.

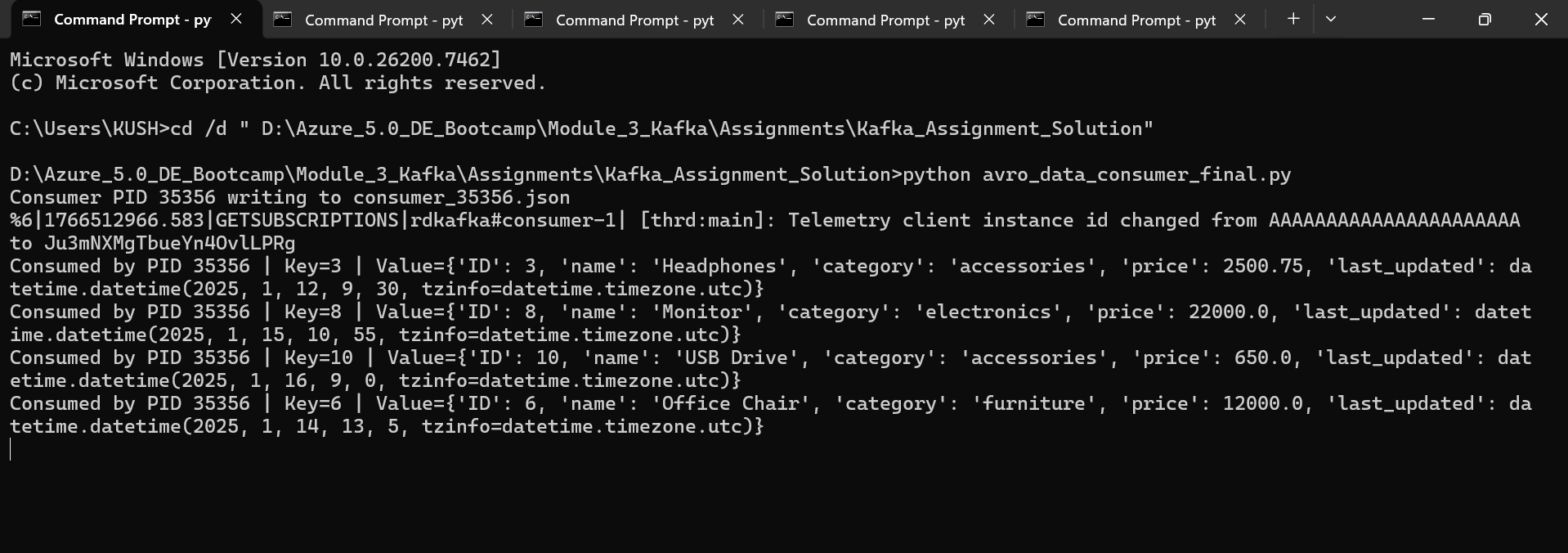
Producer screenshot :

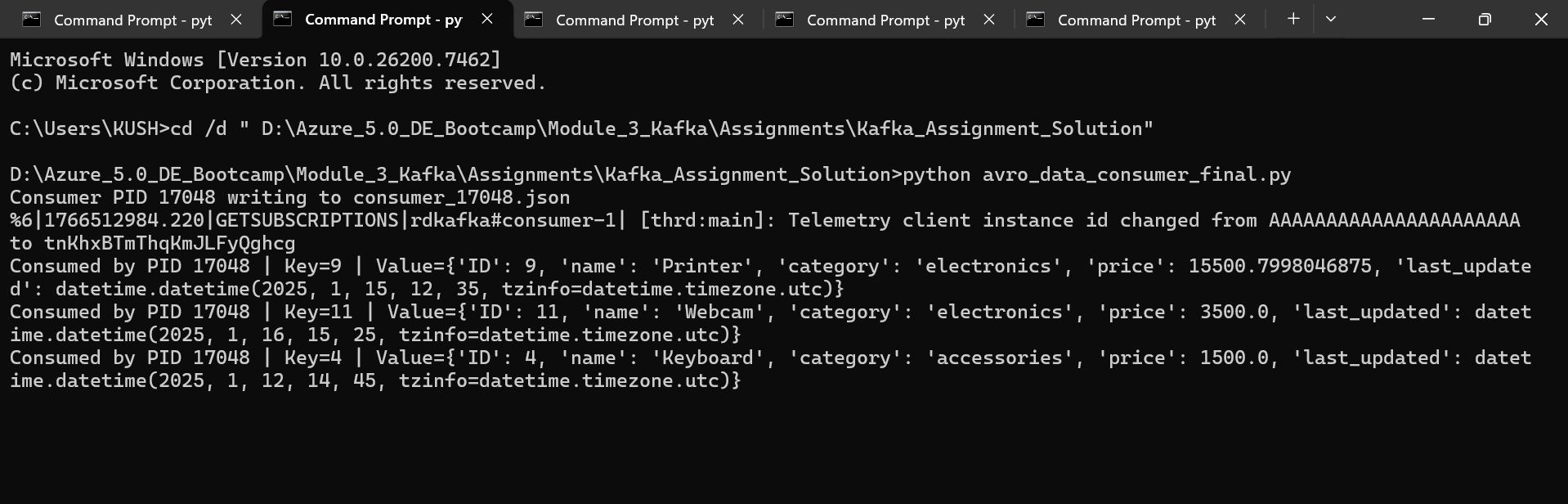


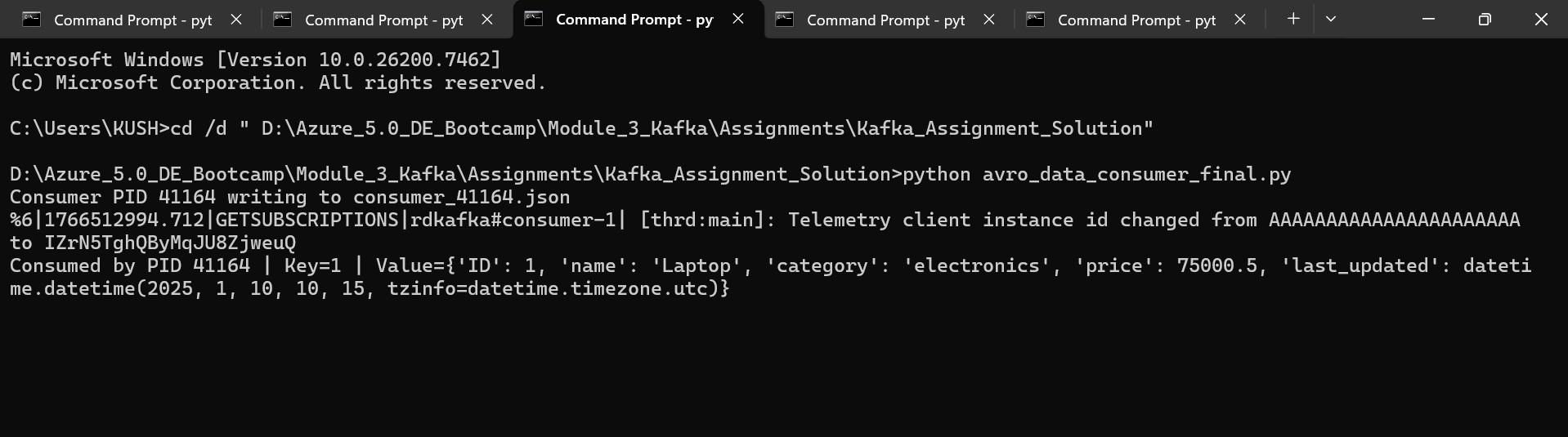
Record processed to producer and published to kafka successfully.

Now as soon as all records are loaded to producer, consumers should start parallely to consume all records.

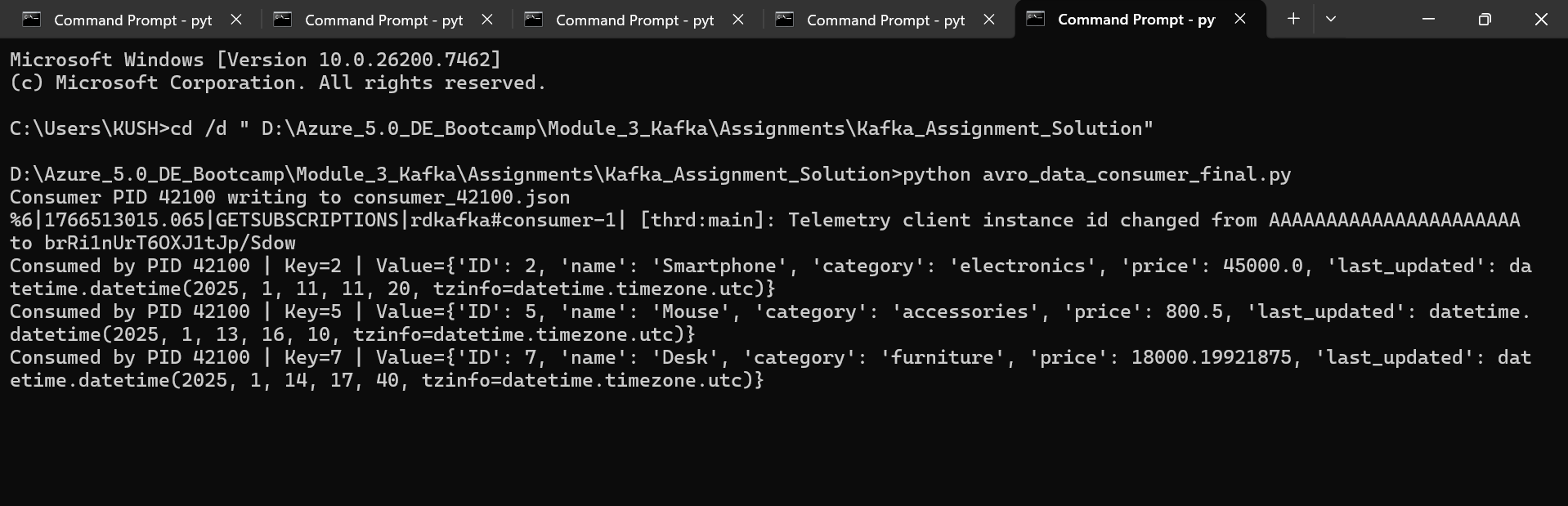
Consumer screenshots :











Consumers start consuming the 12 records ( screenshots attached for consumer 1 to 5 consuming all 12 records in the table initially)

step 4. Load more records in the mysql table and run the producer.

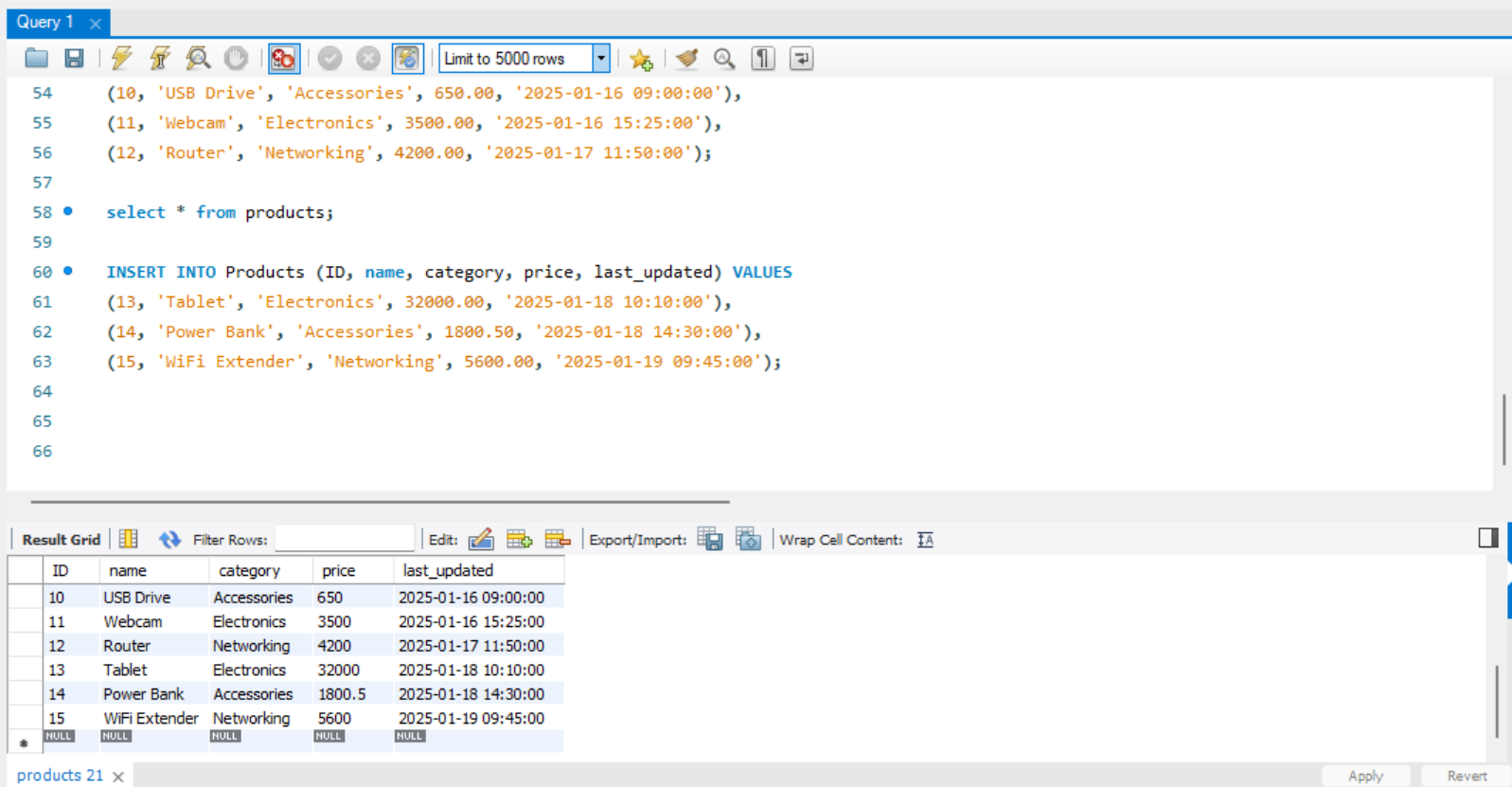
INSERT INTO Product (ID, name, category, price, last\_updated) VALUES

(13, 'Tablet', 'Electronics', 32000.00, '2025-01-18 10:10:00'),

(14, 'Power Bank', 'Accessories', 1800.50, '2025-01-18 14:30:00'),

(15, 'WiFi Extender', 'Networking', 5600.00, '2025-01-19 09:45:00');

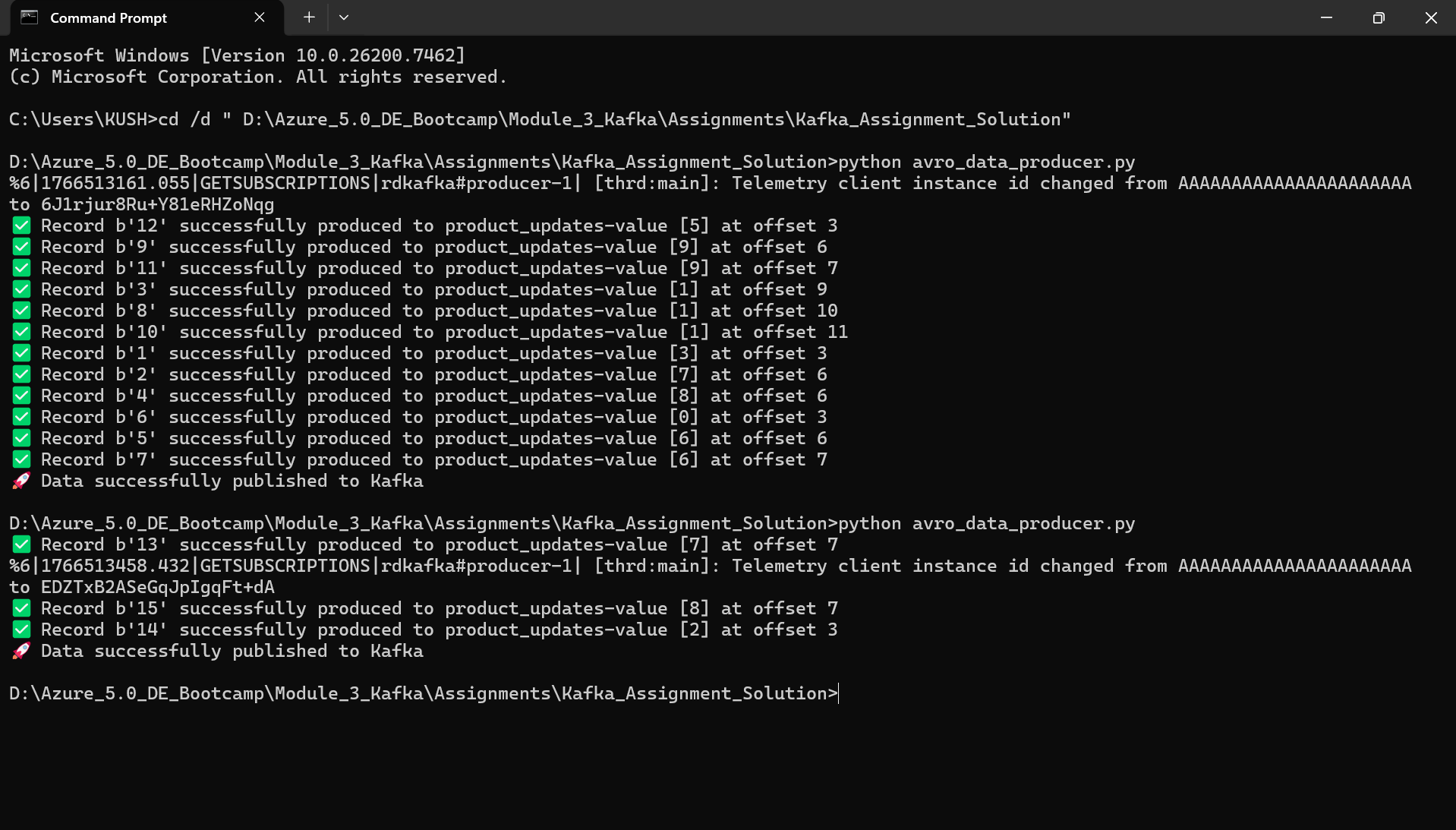
Mysql screenshot :



Again loaded 3 records in the table.

Now producer should pick the records loaded after the records already were present in the table before the first iteration and these records must be picked up by the consumers automatically.

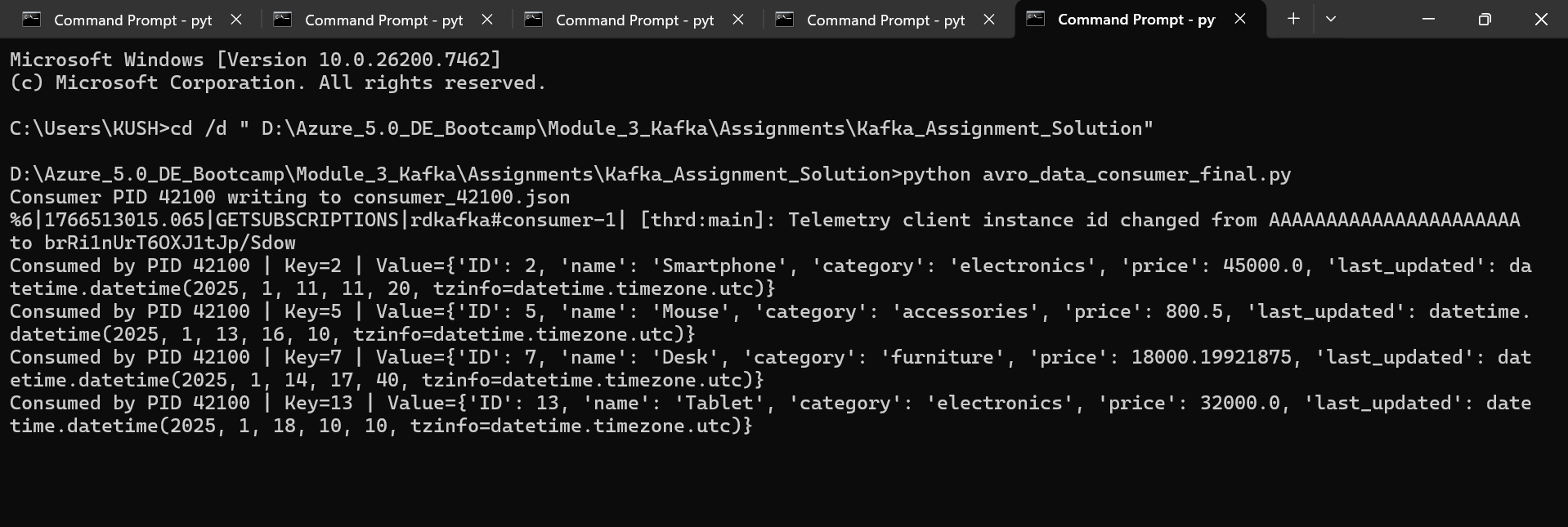
Producer screenshot :



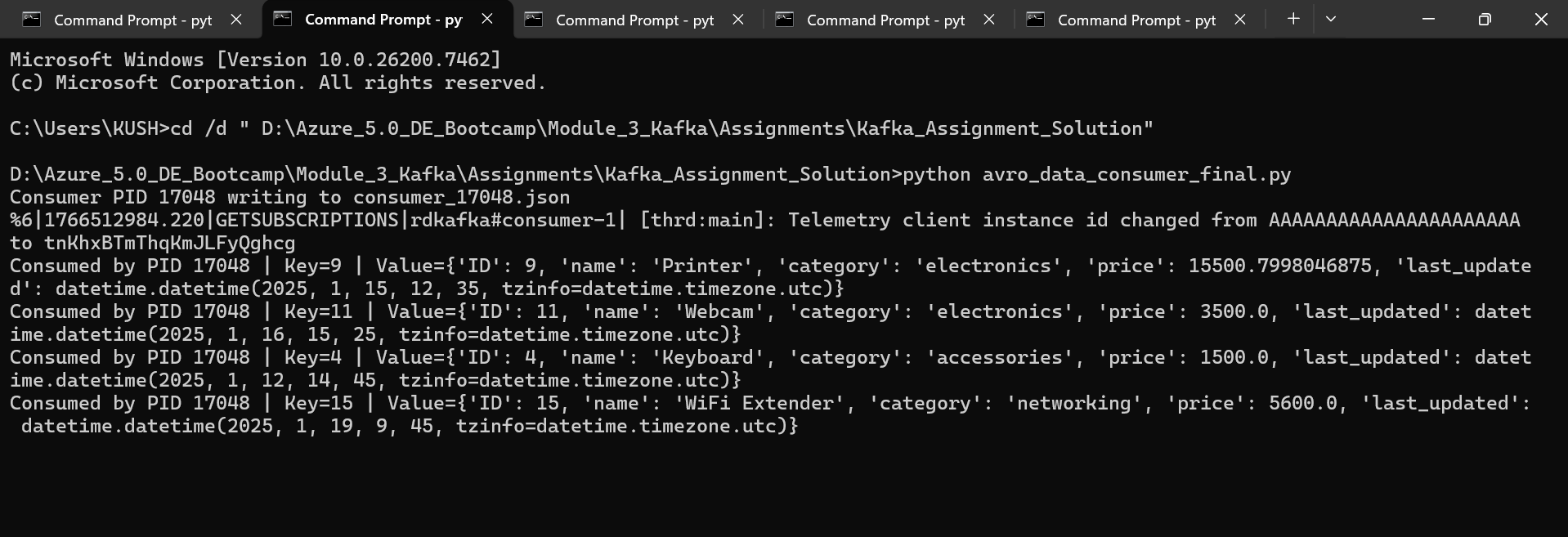
new record processed to producer successfully.

Consumer screenshots :

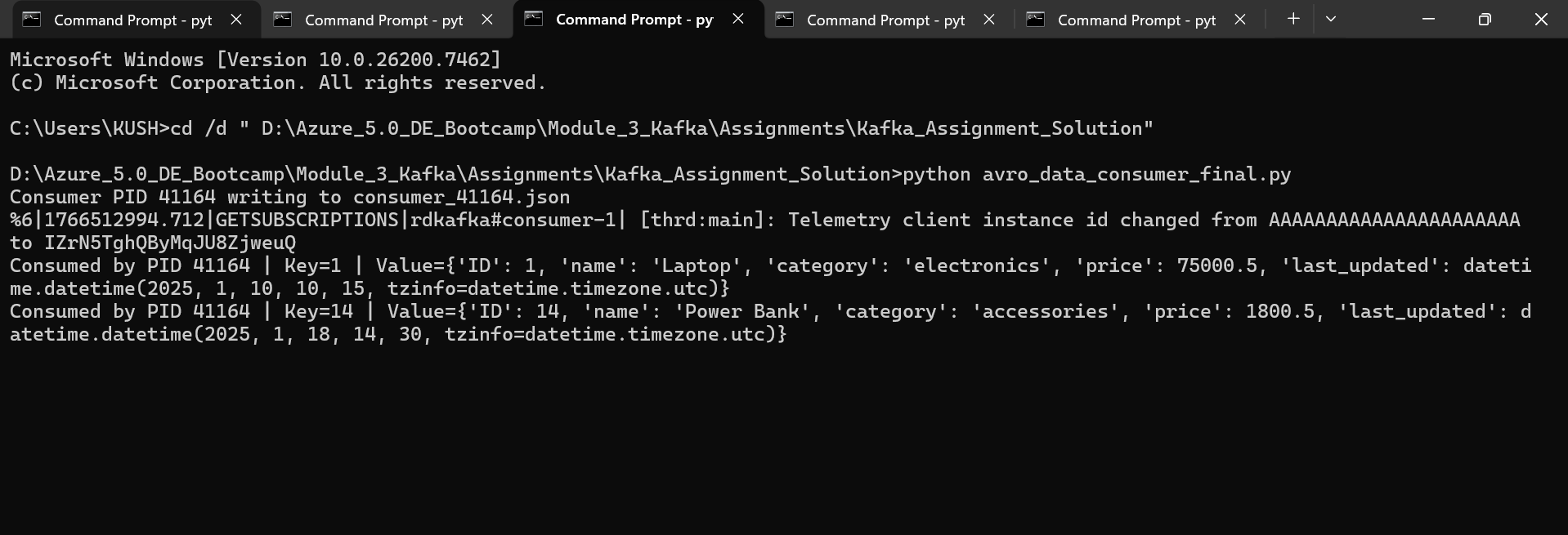
‘Tablet’ consumed:



‘Wifi Extender’ consumed:

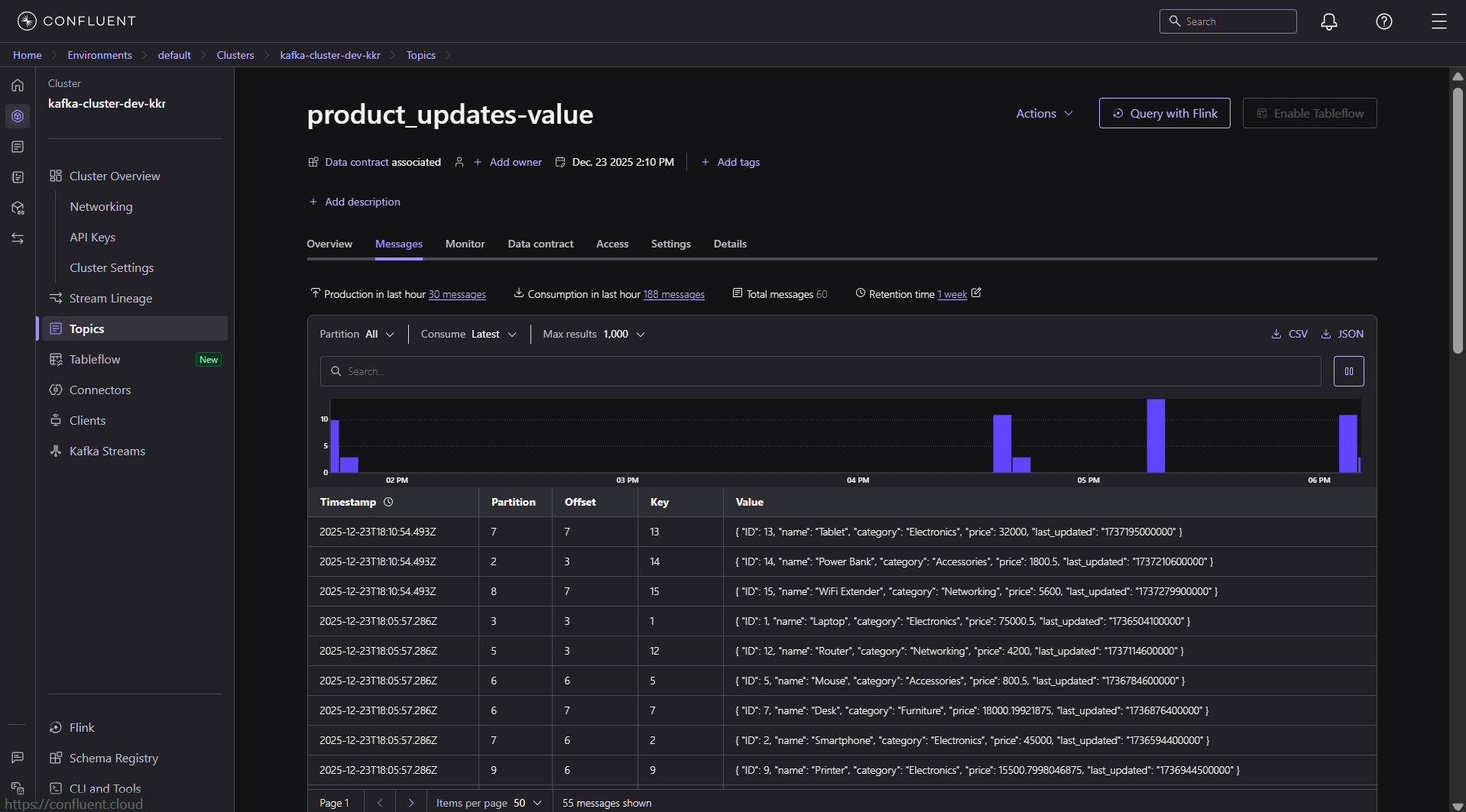


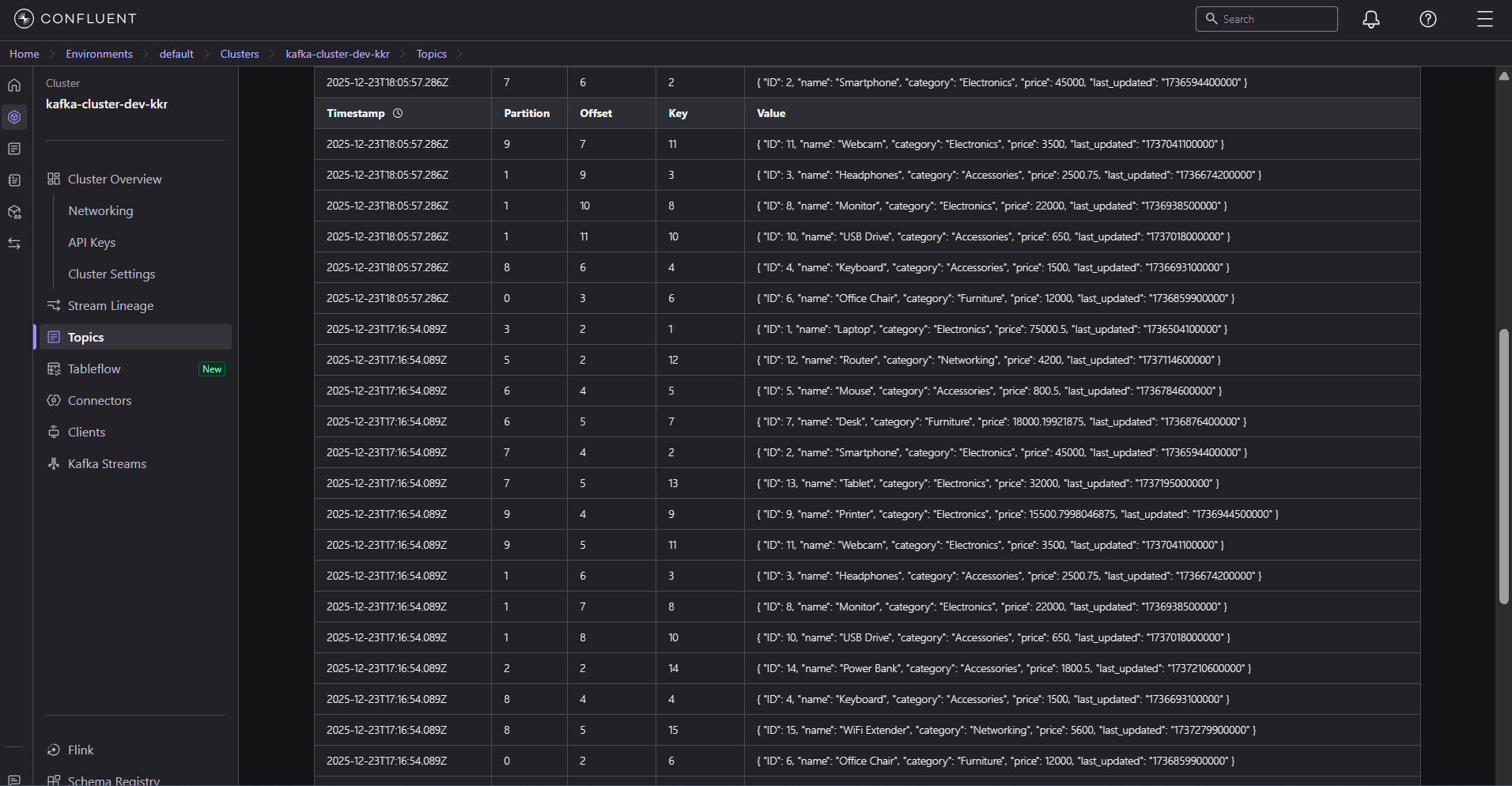
‘Power Bank’ consumed:

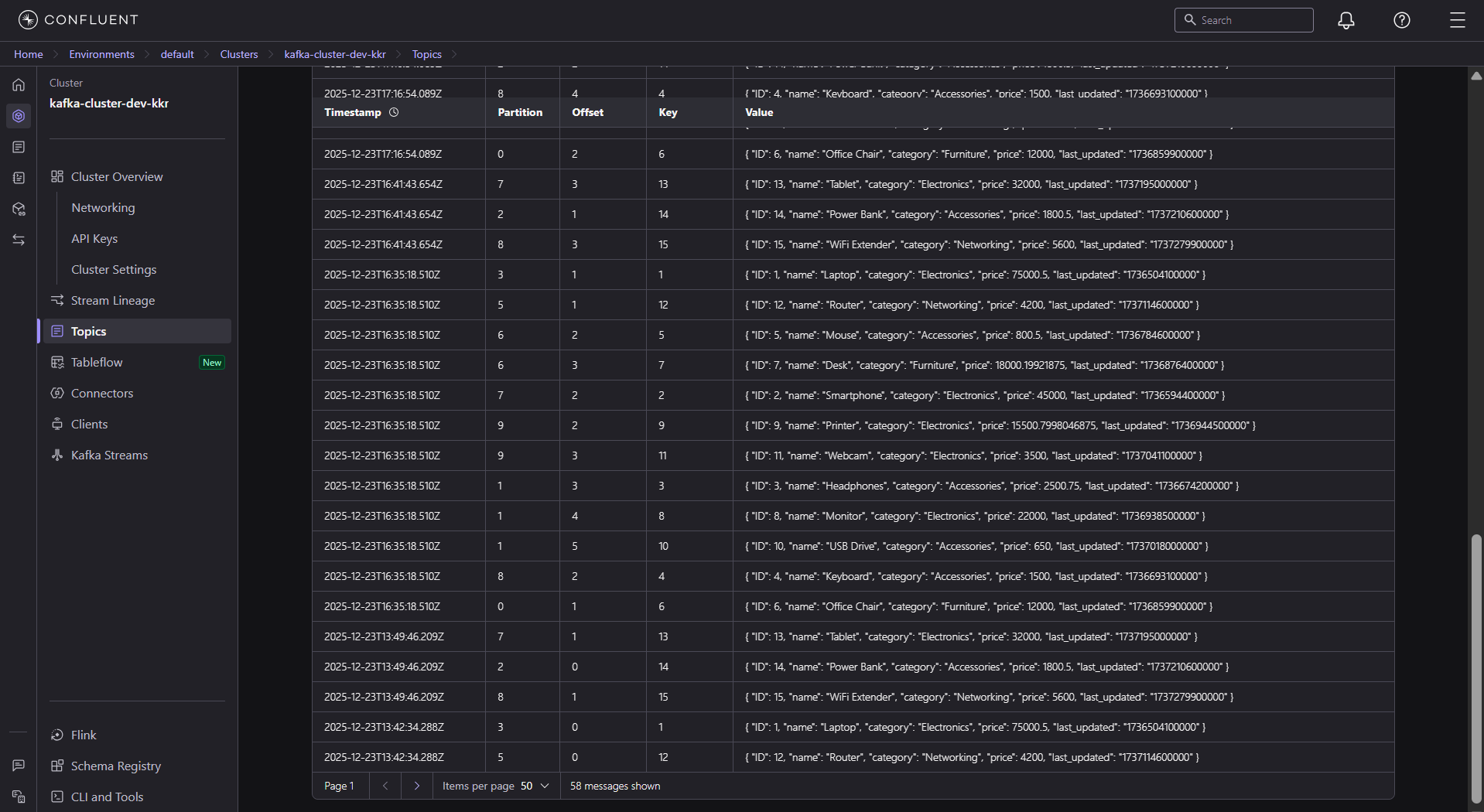


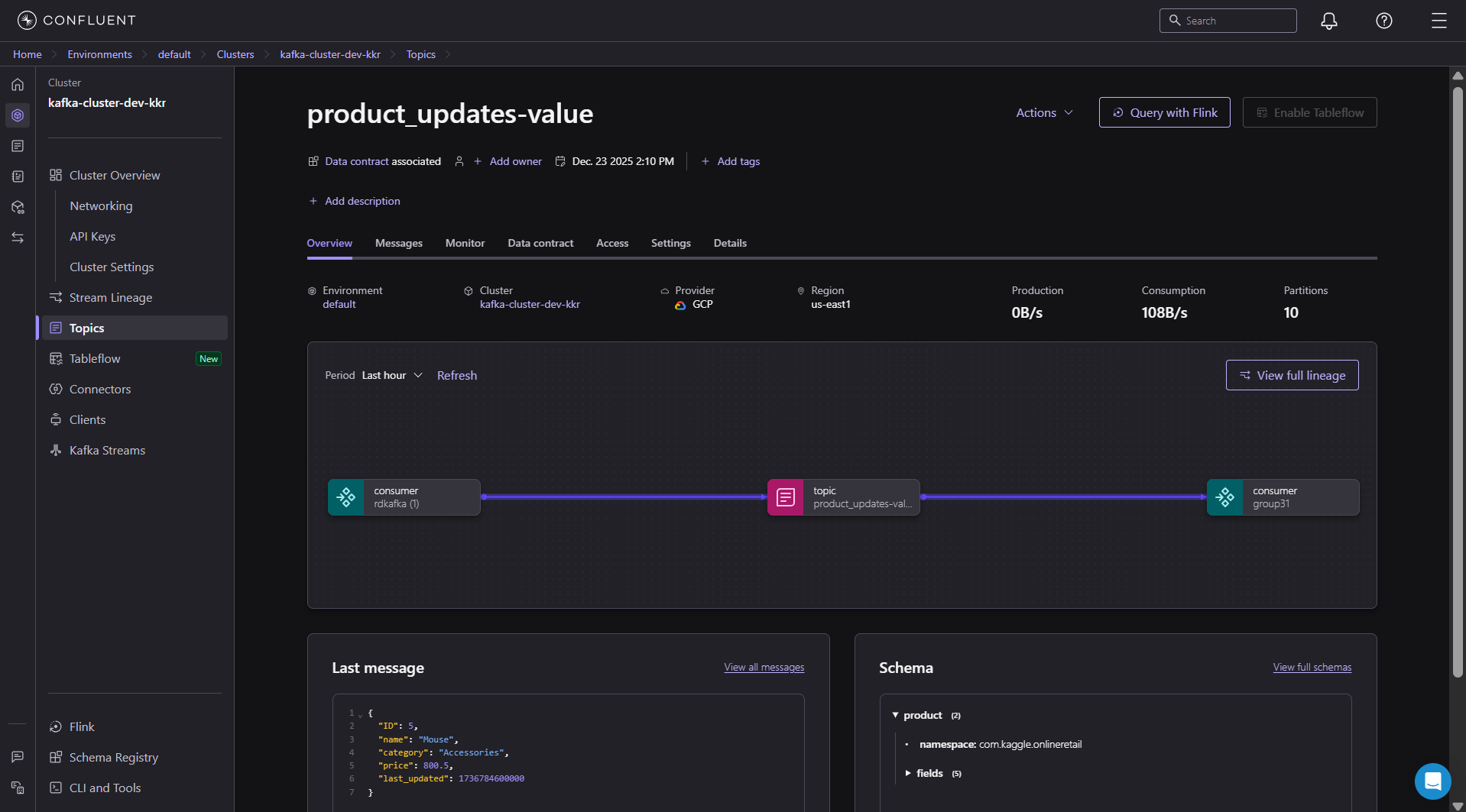
All 3 (with ID as 13,14,15) records consumed by consumers 5,2 and 3 respectively.

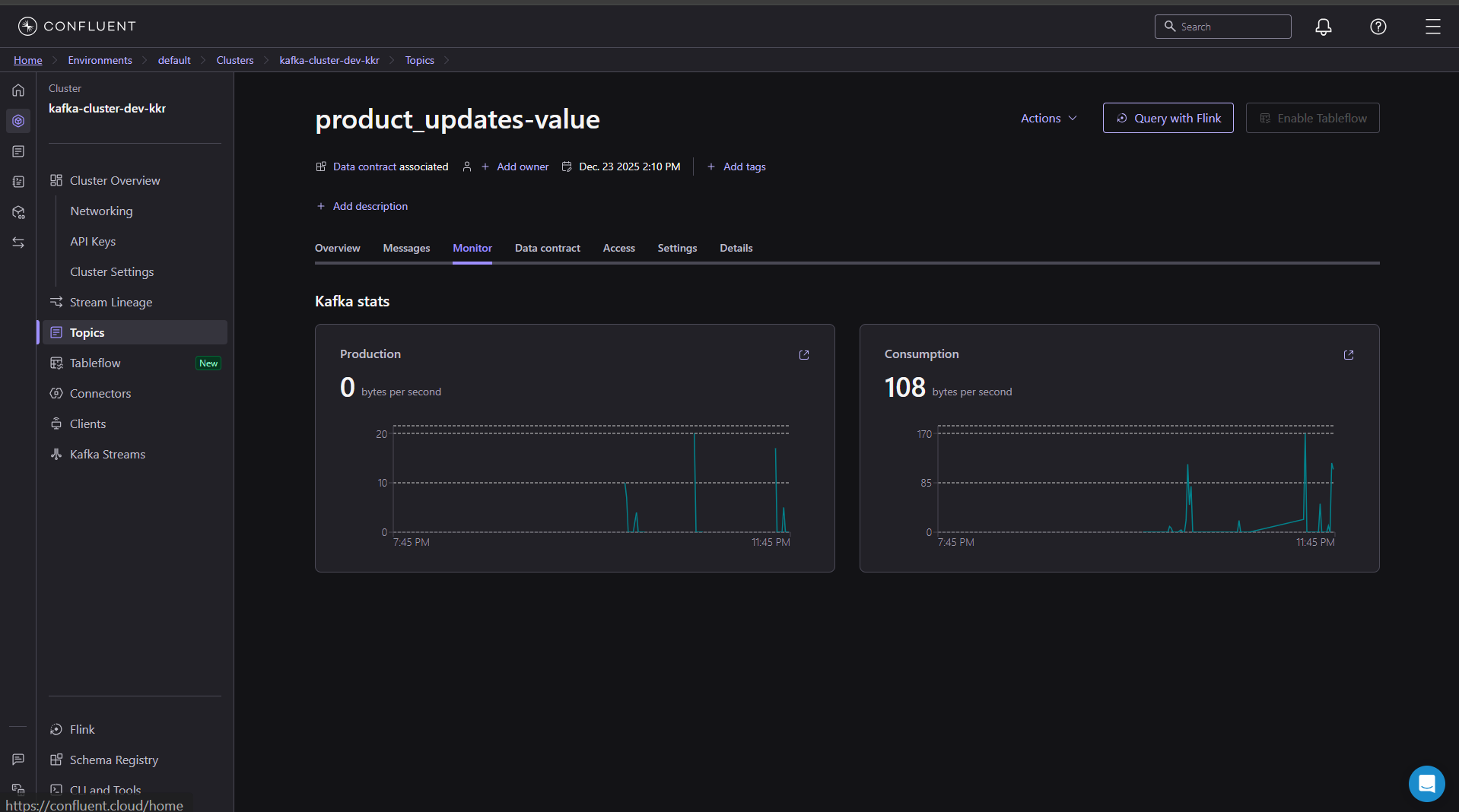
UI screenshots :



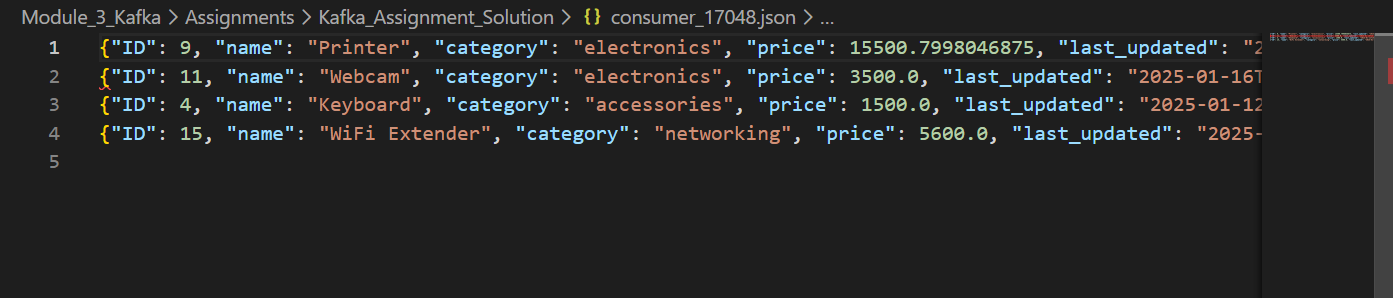


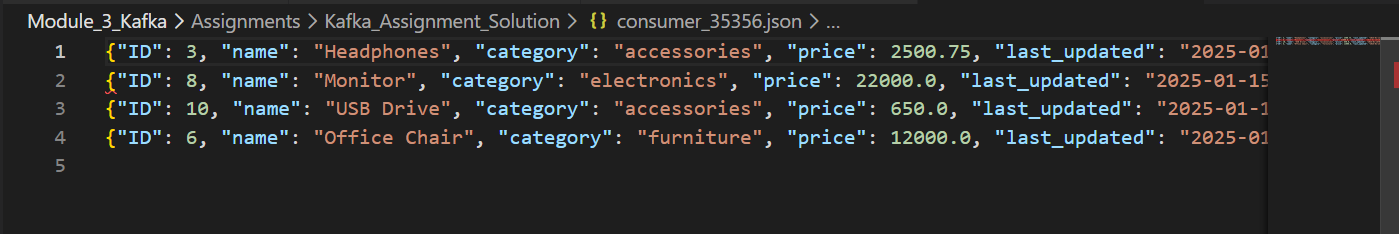


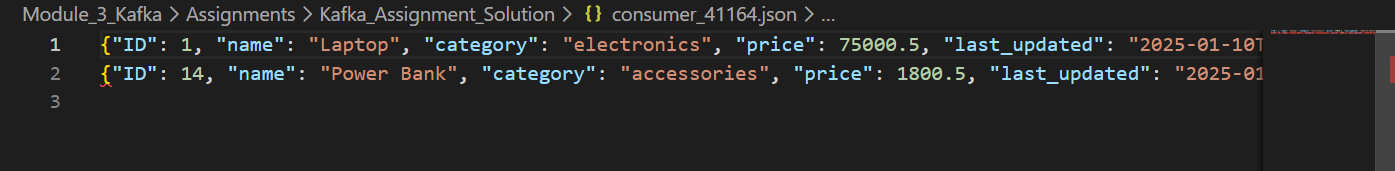


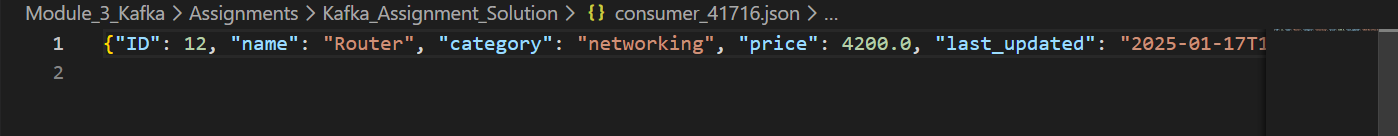


JSON files:



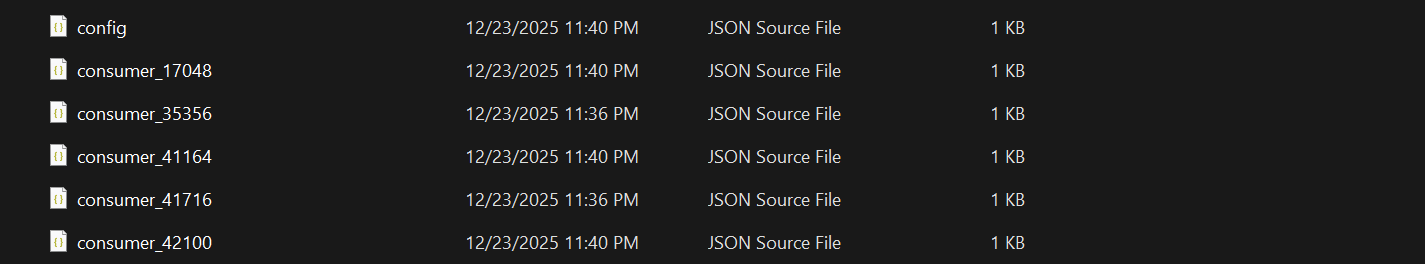








Files created and data is there as expected:



------------------------------------------------------------END------------------------------------------------------------------------------------------------------------------------------------------------END-----------------------------------------------------