Step 1: Snowflake Table Setup (SQL) - This is where all the streamed Kafka messages will land in Snowflake.

CREATE TABLE MY\_TOPIC\_DATA (

DATA VARIANT,

METADATA TIMESTAMP\_LTZ DEFAULT CURRENT\_TIMESTAMP()

);

2- Docker Compose File – docker-compose.yml

Here's the **Docker setup** to launch Kafka, Zookeeper, and Kafka Connect — all needed to stream data.

Place the docker-compose.yml in a project directory

Has Docker installed and running

Just run:

docker-compose up -d

This spins up everything Kafka-related and makes Kafka Connect available on port 8083.

Then **all required services will be automatically set up and run:**

* Kafka
* Zookeeper
* Kafka Connect

**3- Python Kafka Producer to Stream churn\_dataset.csv**

This Python script will:

* Read churn\_dataset.csv
* Convert each row into JSON
* Send it to Kafka topic my\_kafka\_topic

**4- Configure Snowflake Kafka Connector**

This file will tell Kafka Connect **how to forward data from Kafka to Snowflake**.

Just change this to your account –

snowflake.url.name

snowflake.user.name

snowflake.private.key

Make sure your actual snowflake.private.key is base64-encoded and pasted as one continuous line.

Place this JSON file in the working directory

Run this command to submit the connector

Code:

**curl -X POST -H "Content-Type: application/json" \**

**--data @snowflake-connector.json \**

[**http://localhost:8083/connectors**](http://localhost:8083/connectors)

**5-**

**Verify Data in Snowflake**

**Once Kafka is streaming rows from churn\_dataset.csv into my\_kafka\_topic and the Snowflake Kafka Connector is running correctly, Snowflake will begin receiving the data.**

**In Snowflake:**

**Run this SQL to see the streamed records:**

**SELECT \* FROM MY\_TOPIC\_DATA ORDER BY METADATA DESC;**

**You’ll see a DATA column (type: VARIANT) that contains the full JSON row from the CSV.**

**Example row inside DATA:**

**{**

**"customerID": "7590-VHVEG",**

**"gender": "Female",**

**"SeniorCitizen": "0",**

**"Partner": "Yes",**

**...**

**"Churn": "No"**

**}**

**Final Expected Outcome**

* **Kafka receives structured customer rows (21 features) from the CSV**
* **Snowflake connector picks them up automatically**
* **Snowflake table MY\_TOPIC\_DATA stores them as JSON in DATA column**
* **can now plug this into a dashboard or model**

### Overall Project Flow **Prepared Input Dataset**

* Used churn\_dataset.csv (21 features) as the source.
* Each row represents a customer with demographic + usage + contract info.

### **2. Docker-Based Kafka Environment Setup**

* Used docker-compose.yml to spin up:
  + Kafka
  + Zookeeper
  + Kafka Connect
* Ensured all services were running:
  + Zookeeper on port 2181
  + Kafka on port 9092
  + Kafka Connect on port 8083

3. Created Snowflake Table

store each Kafka message in the DATA column as JSON.

### **4. Wrote Python Kafka Producer**

* Read rows from churn\_dataset.csv
* Sent each row as a JSON message to the topic: my\_kafka\_topic
* Used kafka-python library to connect to local Kafka broker.
* Simulates real-time data feed.

### **5. Configured Snowflake Sink Connector**

* Created snowflake-connector.json with fields like:
  + connector.class
  + snowflake.user.name
  + snowflake.private.key
  + snowflake.url.name
  + snowflake.topic2table.map
* Purpose: automatically push messages from Kafka to Snowflake.

I tried setting up the Kafka-to-Snowflake streaming using the connector. I generated a valid PKCS8 RSA key, encoded it properly, and added it to the config JSON. I also verified my Snowflake URL, user, and table. But every time I posted the connector, it threw snowflake.private.key must be a valid PEM RSA private key or connection errors. I tried all known formats, encoding methods, and even validated via /config/validate but nothing worked. Despite following all steps correctly, I couldn’t get the connector to register. I’m now sending you all files from start to end. Please execute them — this is my final request after trying everything.