**Kafka Assignment**

Get the resources from here:

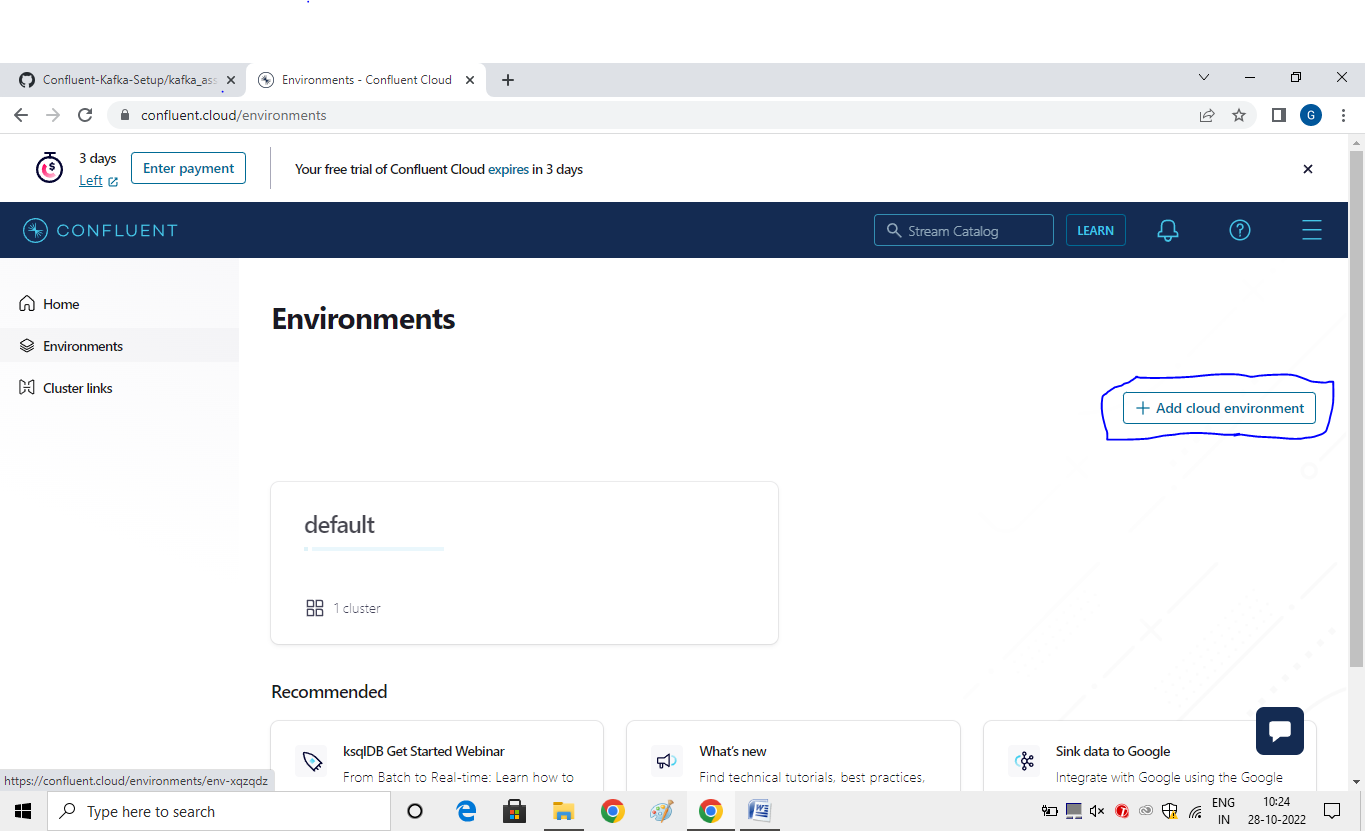
Download Data Link -> <https://github.com/shashank-mishra219/Confluent-Kafka-Setup/blob/main/restaurant_orders.csv>

1. Setup Confluent Kafka Account

<https://confluent.cloud/login> Click link and create confluent Kafka account.

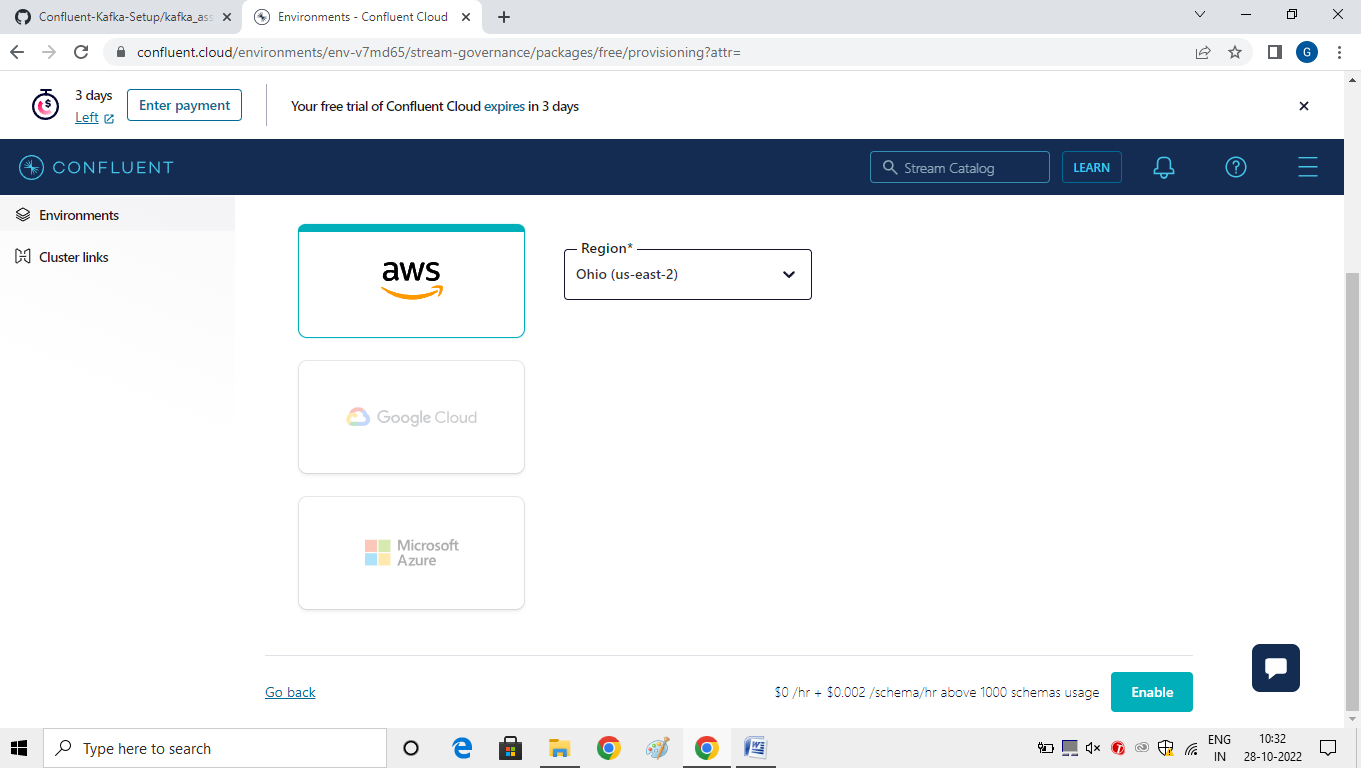
1. Create one Kafka topic named as "restaurant-take-away-data" with 3 partitions.

Step 1:



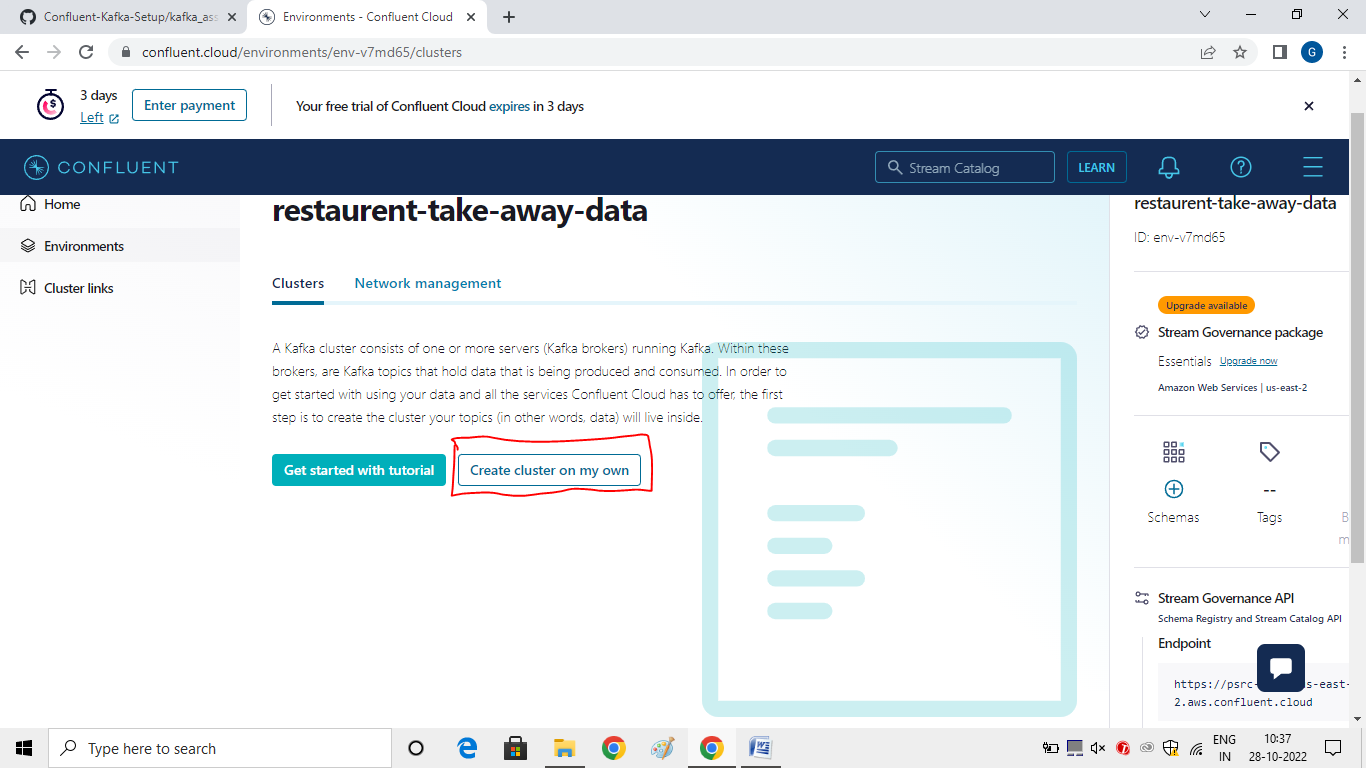
Step 2: Enter Environment name as “restaurent”

And Click on Begin Configuration as free. Choose any region

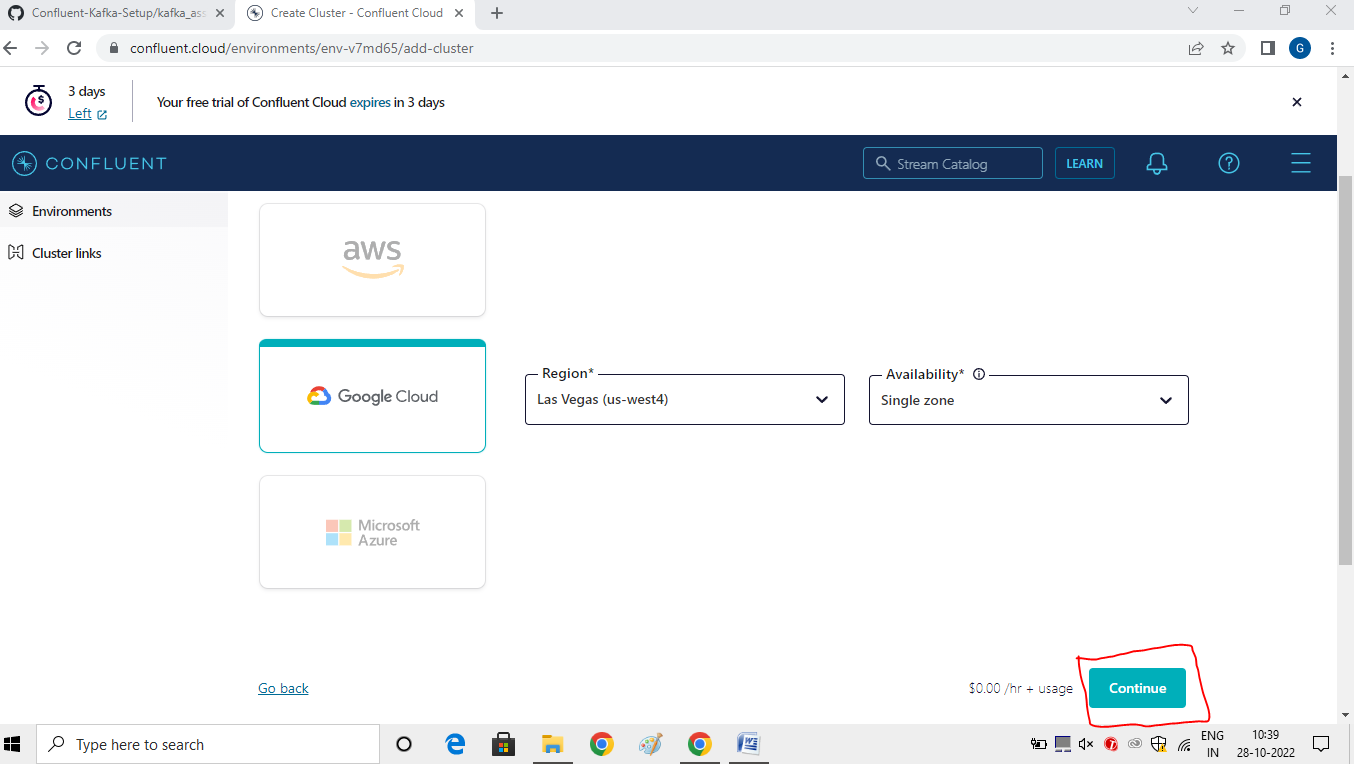


Click to enable .

Step 3:



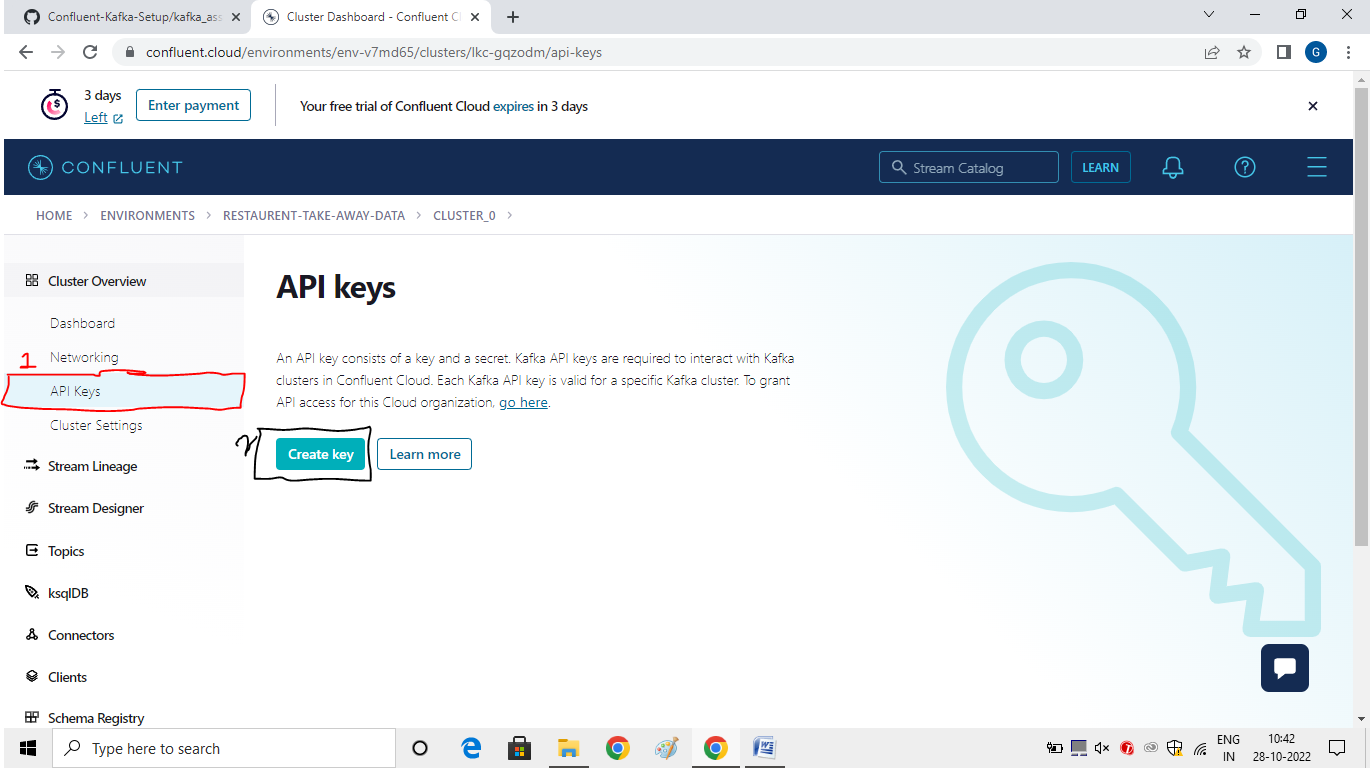
After that click on “Begin Configuration “ Basic free. And click on “Continue”



Skip the Payment Method. After that click on “Launch Cluster”.

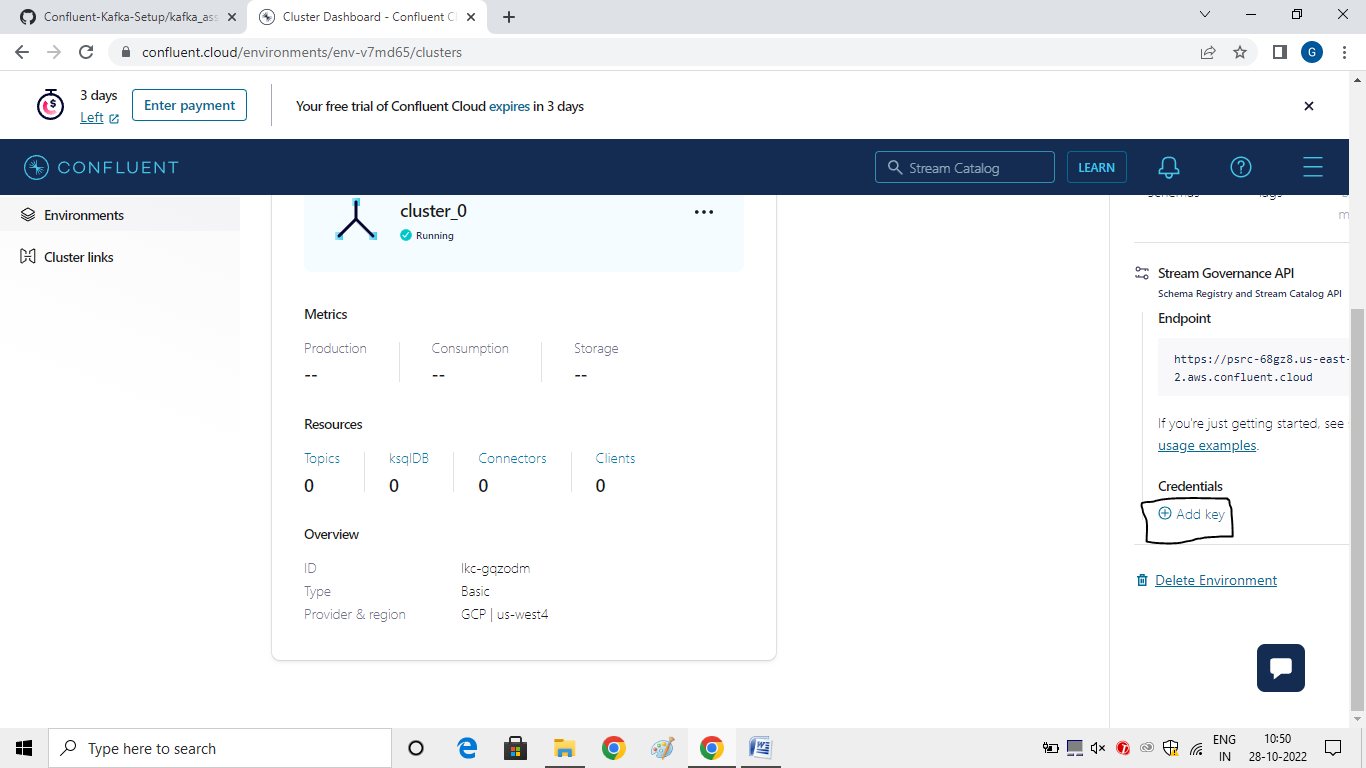
Step 4:

Click on API KEY . And Create key.



Step 4:

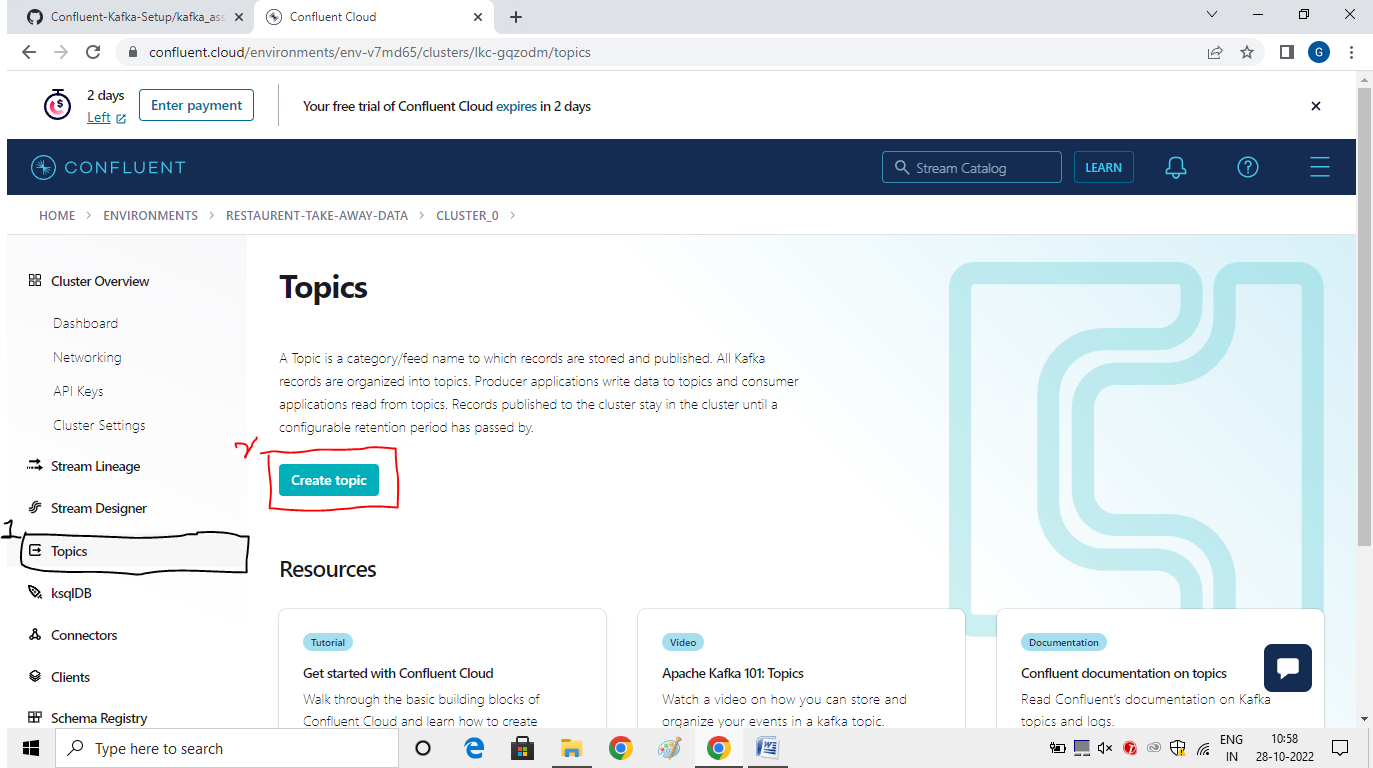
Add Key-> Create Key-> Enter Schema Registry Description and download.



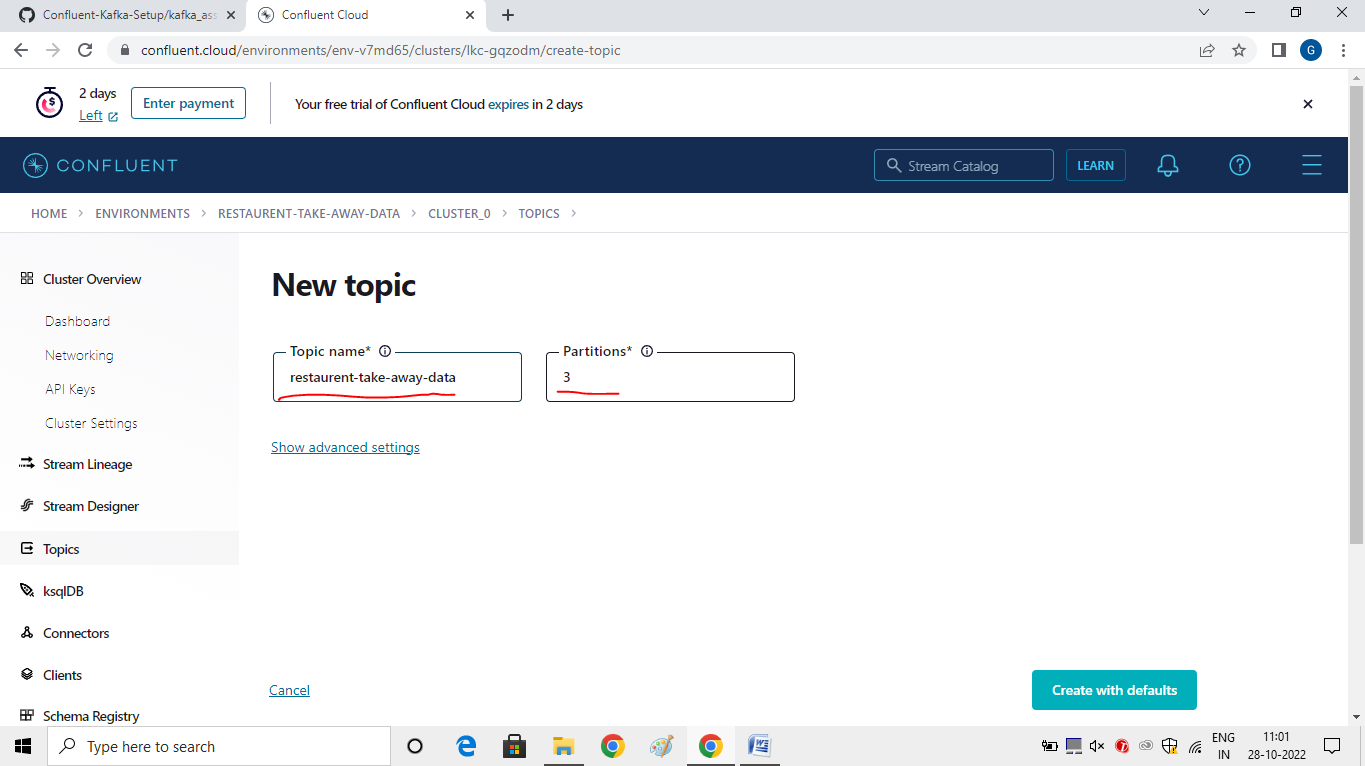
1. Setup key (string) & value (json) schema in the confluent schema registry

Step 5: Create topic

Click on “Topics” -> Create Topic

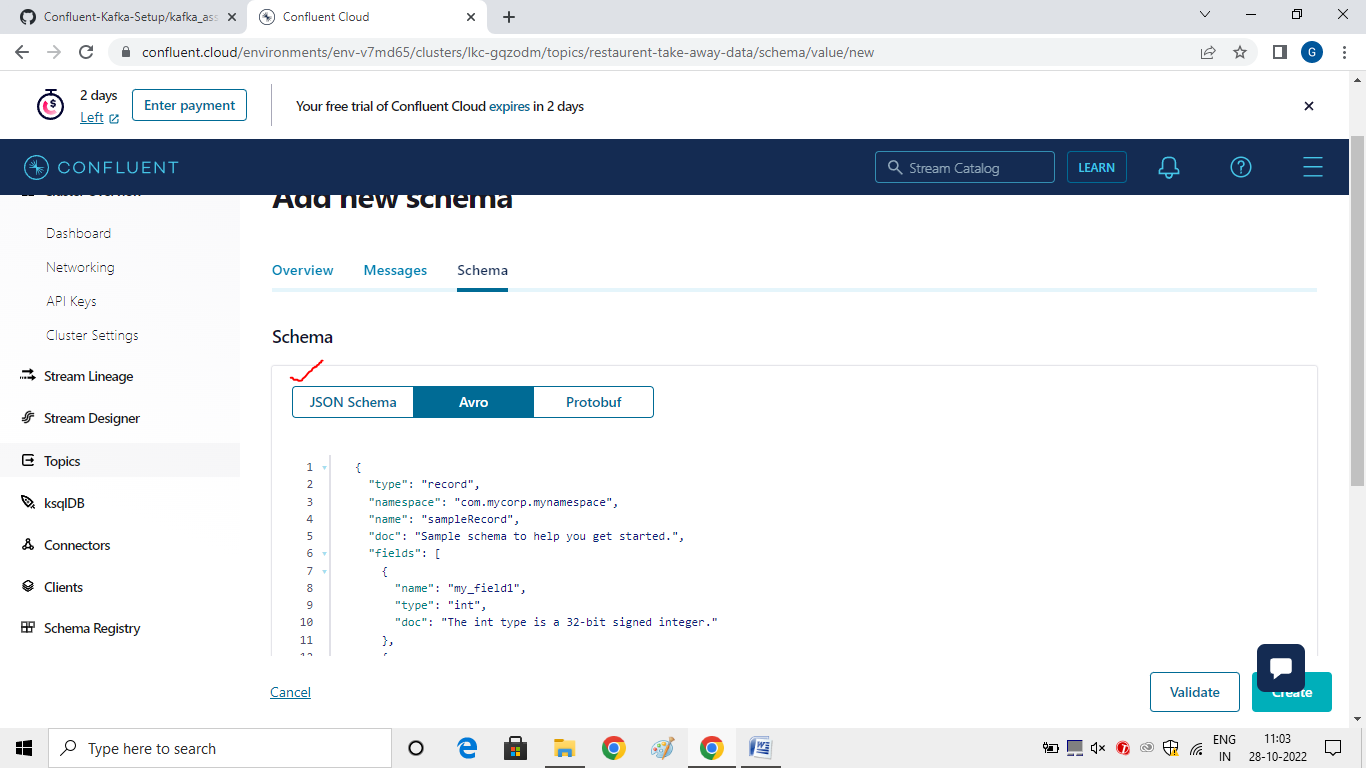


Enter Topic Name and Partitions.



Click on “create with Defaults”

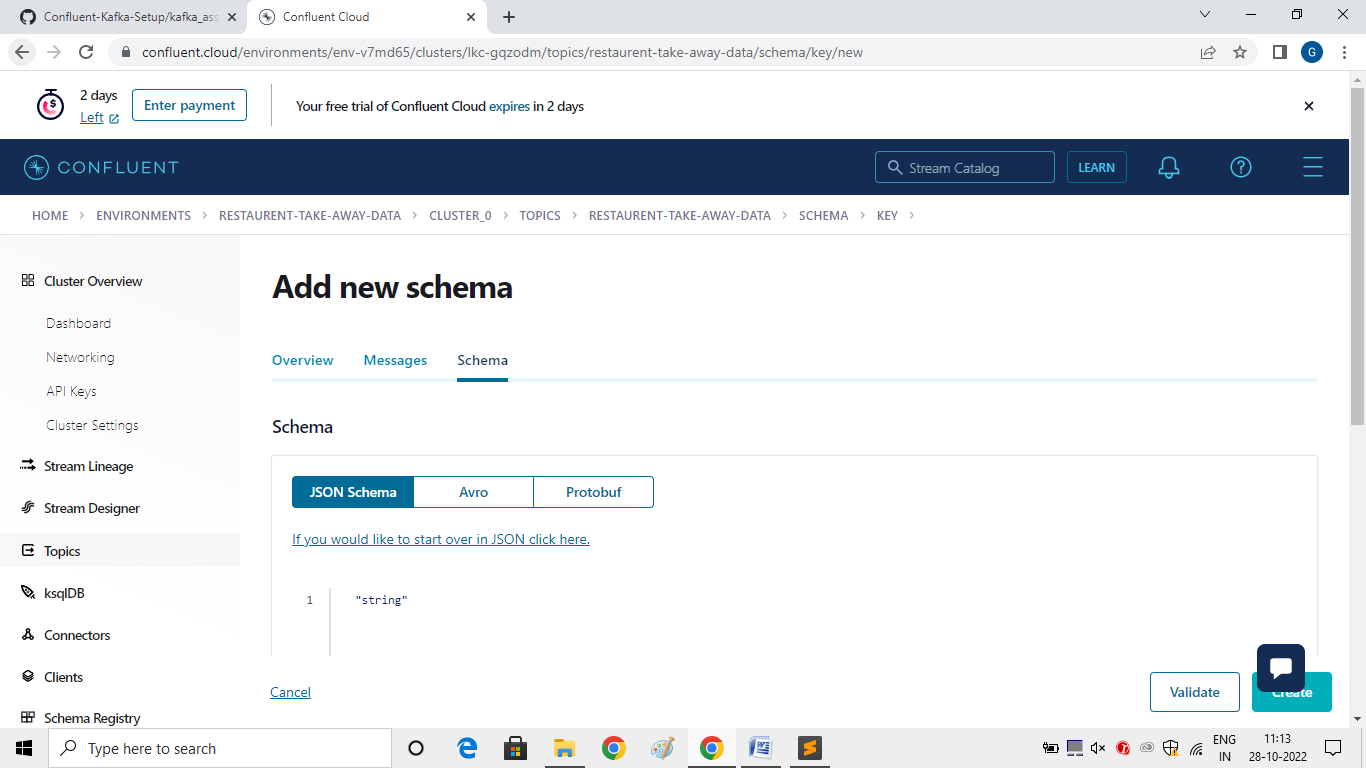
Step 5: Set up Schema



Click on “Json schema” -> Enter the schema -> Click on “validate”

The validation will be passed click on “Create”

After that click on “Key” -> Enter schema is “string”



Click on Create.

|  |
| --- |
|  |

1. Write a kafka producer program (python or any other language) to read data records from restaurent data csv file, make sure schema is not hardcoded in the producer code, read the latest version of schema and schema\_str from schema registry and use it for data serialization.

Run the Producer Code in python or java Languages.

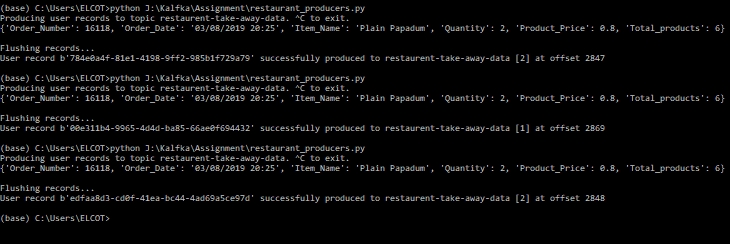
Refere the Producer Code here:

<https://github.com/Gunasekar13/Kafka/blob/main/iNeuron_Assignment/restaurant_producers.py>

1. From producer code, publish data in Kafka Topic one by one and use dynamic key while publishing the records into the Kafka Topic



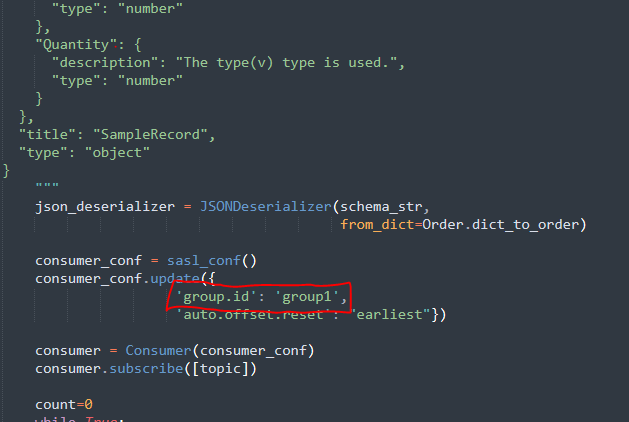
1. Write kafka consumer code and create two copies of same consumer code and save it with different names (kafka\_consumer\_1.py & kafka\_consumer\_2.py), again make sure lates schema version and schema\_str is not hardcoded in the consumer code, read it automatically from the schema registry to desrialize the data. Now test two scenarios with your consumer code:



1. Use "group.id" property in consumer config for both consumers and mention different group\_ids in kafka\_consumer\_1.py & kafka\_consumer\_2.py, apply "earliest" offset property in both consumers and run these two consumers from two different terminals. Calculate how many records each consumer consumed and printed on the terminal.

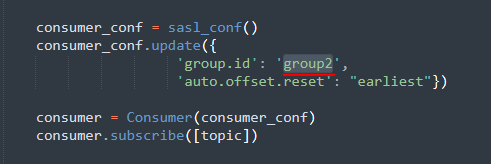
Consumer1 code here:

<https://github.com/Gunasekar13/Kafka/blob/main/iNeuron_Assignment/restaurant_consumer1.py>

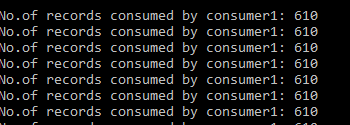


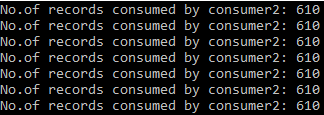
Consumer2 Code here:

<https://github.com/Gunasekar13/Kafka/blob/main/iNeuron_Assignment/restaurant_consumer2.py>



Outputs:





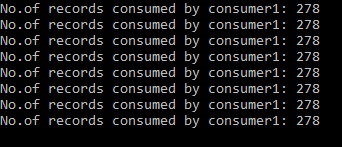
1. Use "group.id" property in consumer config for both consumers and mention same group\_ids in kafka\_consumer\_1.py & kafka\_consumer\_2.py, apply "earliest" offset property in both consumers and run these two consumers from two different terminals. Calculate how many records each consumer consumed and printed on the terminal.

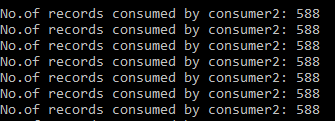
Make Sure the group id‘s are same in Consumer1 and consumer2 code

[https://github.com/Gunasekar13/Kafka/blob/main/iNeuron\_Assignment/restaurant\_consumer1.py](https://github.com/Gunasekar13/Kafka_Assignmen_iNeuron/blob/main/restaurant_consumer1.py)

<https://github.com/Gunasekar13/Kafka/blob/main/iNeuron_Assignment/restaurant_consumer2.py>

Outputs:





1. Once above questions are done, write another kafka consumer to read data from kafka topic and from the consumer code create one csv file "output.csv" and append consumed records output.csv file.

<https://github.com/Gunasekar13/Kafka/blob/main/iNeuron_Assignment/restaurant_consumer.py>

