# Example written based on the official

# Confluent Kakfa Get started guide https://github.com/confluentinc/examples/blob/7.1.1-post/clients/cloud/python/consumer.py

from confluent\_kafka import Consumer

import json

import ccloud\_lib

import time

# Initialize configurations from "python.config" file

CONF = ccloud\_lib.read\_ccloud\_config("python.config")

TOPIC = "my\_first\_topic"

# Create Consumer instance

# 'auto.offset.reset=earliest' to start reading from the beginning of the

# topic if no committed offsets exist

consumer\_conf = ccloud\_lib.pop\_schema\_registry\_params\_from\_config(CONF)

consumer\_conf['group.id'] = 'my\_weather\_consumer'

consumer\_conf['auto.offset.reset'] = 'earliest' # This means that you will consume latest messages that your script haven't consumed yet!

consumer = Consumer(consumer\_conf)

# Subscribe to topic

consumer.subscribe([TOPIC])

# Process messages

try:

while True:

msg = consumer.poll(1.0) # Search for all non-consumed events. It times out after 1 second

if msg is None:

# No message available within timeout.

# Initial message consumption may take up to

# `session.timeout.ms` for the consumer group to

# rebalance and start consuming

print("Waiting for message or event/error in poll()")

continue

elif msg.error():

print('error: {}'.format(msg.error()))

else:

# Check for Kafka message

record\_key = msg.key()

record\_value = msg.value()

data = json.loads(record\_value)

weather = data["degrees\_in\_celsion"]

print(f"It's currently {weather} degrees")

time.sleep(0.5) # Wait half a second

except KeyboardInterrupt:

pass

finally:

# Leave group and commit final offsets

consumer.close()