Code Challenge for the Data Engineering role

Dear Applicant,

you applied for an engineering position in a data intensive environment that is heavily based on Apache Kafka, Kubernetes and AWS. In your daily work you will be confronted with Apache Kafka and are supposed to create and maintain services around it.

Situation

At one day, you noticed that one of your kafka topics contains malformed data. You identified the problem and stopped the wrong ingestion. Now you also need to correct the already wrong data in kafka. Therefore, you must write a simple application which reads the wrong data, fixes it and produces it into a new topic.

Challenge

1. Setup a local Kafka cluster for development.
2. Prepare two Kafka topics.

   One topic is called "input_topic".

   The other topic is called "output_topic".

3. Populate the "input_topic" with five Kafka messages in JSON format. The Kafka messages are provided below.

   ```
   {"myKey": 1, "myTimestamp": "2022-03-01T09:11:04+01:00"}
   {"myKey": 2, "myTimestamp": "2022-03-01T09:12:08+01:00"}
   {"myKey": 3, "myTimestamp": "2022-03-01T09:13:12+01:00"}
   {"myKey": 4, "myTimestamp": ""}
   {"myKey": 5, "myTimestamp": "2022-03-01T09:14:05+01:00"}
   ```

   Unfortunately, all the Kafka messages are malformed, the field "myTimestamp" is written in the wrong time zone Europe/Berlin. However, all timestamps (if available) should be by default in UTC.

   Example:

   *Wrong: 2022-03-01T**09**:11:04+01:00 -> Europe/Berlin*

   *Correct: 2022-03-01T**08**:11:04+00:00 -> UTC*

4. Write a simple transformer application in **python** to consume all the messages from "input_topic" and write the corrected messages to "output_topic".
5. Dockerize your application, so it can be run with Docker.
6. Publish your code to a public git repo, e.g. GitHub.
7. Add a readme.md that describes all steps in a way that we can reproduce your solution on our workstations. This should include the commands how to install and start the local Kafka setup, how to insert the test data and how to run and stop your application.
8. Share with us the link to your public git repo.

Notes/Hints:

- Depending on your experience we expect you to finish this challenge within 4 to 8 working hours. You have 7 calendar days in total to finish the challenge and send us the link to the public git repo.
- It's not required to setup a Kafka cluster with more than 1 broker. Keep it simple. It's only important to provide a local Kafka setup that your application can connect to and use. We don't want to evaluate your Kafka admin skills, rather want to see how you can create a local environment for unit tests or development purposes.
- Try to run your kafka cluster in a container orchestration framework (e.g. minikube or docker-compose). If you run into time issues of course you can also install kafka on your local host.
- Filling the input topic can be done by simple kafka-cli commands.