Data Engineer Technical Test

QUESTION 1: Python programming

Implement a software that can read a dataset from a specific file path, apply some transformations to its fields and write the result into another path.

Requirements:

- Datasets can be read and written in CSV and at least another format: Parquet or JSONL (JSON Lines).
- The program has a library of transformations that can potentially be applied to the dataset, but on each execution we select which ones we want to apply to which columns.
- Anyone without programming skills should be able to execute the script. To do that, the user will provide a JSON configuration file. Your solution has to parse this configuration and generate the result based on it.
- Your program must be packed as a Docker image to be easily deployed and executed.
- Test Driven Development will be highly valued.
- The transformations that you need to implement are:
    - **birthdate_to_age:** Given a date, it computes the age of a person and creates a new column with the result.
    - **hot_encoding:** Given a categorical column with n possible values, it replaces it with n binary columns. For example, given the column “color” with 3 possible values: “blue”, “red” and “green”, we will create 3 columns: “is_blue”, “is_red” and “is_green”, that will be 1 or 0 depending of the value of the original column.
    - **fill_empty_values:** It replaces the empty values of a column with another value. The value to be replaced with is passed as a parameter and it can be a constant or one of the following keywords: “mean”, “median” or “mode”.
      If we pass one of these keywords, we replace the empty values with the mean / median / mode of the rest of the elements of the column.

Example: To test your code, we provide you a small dataset called *bookings.csv* and the json config file (*clean_bookings.json*) that should be passed to your program as an argument to process it and generate a clean one.

QUESTION 2: Architecture

We want to collect telemetry events sent by our cars and store them in a Data Lake.

Design a solution to fulfill these requirements:

- Connected car events will be consumed in real time from Kafka, which is deployed in Kubernetes.
- We are interested in 3 Kafka topics: battery_level, fuel_level and mileage. Events from each topic have a different schema.
- Sometimes we receive an event with incorrect data that does not match this schema. They must be discarded.
- All the events have the field "occurredOn", which is the datetime when they occurred. Due to connectivity issues, it's usual to have a delay between the "occurredOn" and when we can actually consume the event from Kafka.
- It should be easy and fast to retrieve from the Data Lake the events selecting a specific topic and the specific datetime when they occurred (with minute granularity).
- Since there is a high volume of data, the solution must be scalable.

Make a diagram with your high-level architecture proposal to solve this problem. The diagram should show the needed components and the technology that you would choose for each one. You can also include explanations.