Create a small service that consumes and produces messages to a Kafka broker. The service is a ping-pong with a delayed response of 30 seconds, that receives messages like this:

```
{
        “transaction-id”: TRANSACTION-ID,
        “payload”: {
                “message”: “ping”,
                “force_error”: false
        }
}
```

And responds with a message like this:

```
{
        “transaction-id”: TRANSACTION-ID,
        “payload”: {
                “message”: “pong”,
                “processing_time”: TIME
        }
}
```

Where:

- TRANSACTION-ID is a randomly generated UUID
- TIME is the time the service took to generate the response.

The service will respond for success and error processing on different topics. To simulate error conditions, adding a field: force_error: true to the event payload is allowed.
Inbound and outbound topics shall be configurable.
Take into consideration that the service should be idempotent on the success branch, this is:

- If a given message is received more than once, then the service shall respond with the last response without attempting to process the message, but only if the previous time the response was successful
- If a given message is received more than once and the previous attempt to process the message failed, then the service shall try to reprocess the message and respond accordingly.
- If a given message was received and failed more than a configurable amount of times, then the message is not reattempted and sent to a dead letter queue topic. No future request to process the same message shall be considered.

It is expected that you deliver the exercise in a self-contained manner, so it is easy to deploy and test.