Data Engineering
Senior Tech Assessment

Thank you for taking the time to accept this challenge. We hope you enjoy the challenge we have prepared and that it piques your curiosity for the work we do at Loka. We wish to give you the best possible preview of the kind of work we do and the skills needed.

You need to document/explain your code and thought process so we can understand it better and prepare relevant questions to make the discussion more interesting.

Challenge

Introduction

Eventsim is a program that generates event data for testing and demos. It's designed to replicate page requests for a fake music website (picture something like Spotify); the results look like real user data, but are totally fake. For this tech assessment, you are going to consume the event logs from the simulator as a streaming. The data is generated inside a while loop, creating a continuous flow of logs and a fluentd agent is tailing the event logs from the Eventsim generator and sending these as packages of JSON files to a Minio cluster (a compatible S3 storage).

The purpose of this challenge is to automate the build of a simple yet scalable data lake and data warehouse that will enable our BI team to answer questions like:

- What are the locations that used more the website?
- What are the most listened to songs in a period of time?
- How many hours of music a user consumes in a period of time?

Goal

We would like to ask you to develop a solution that:

1. Fetches the streaming data and stores it on a data lake;
2. Create tables with proper data index/partitions to access the data;
3. Create a couple of transformed tables that facilitate analysis by BI teams;
4. Store the transformed data on a data warehouse. The data warehouse should be SQL-queriable (SQL database or a solution like AWS Athena).

You can develop your solution locally but you must provide, at least, a sketch on how you would set up the application on **AWS**. Also, for the data warehouse layer, you should devise a data model.

Data

The data for this challenge is generated using this Eventsim fork.

The current fork sinks the streaming data in Minio, however it is possible to update the fluentd agent output to S3 or another sink.

Technical assumptions

- The fetching process should be a streaming.
- Files on the "raw" S3 bucket can disappear but we might want to process them differently in the future;
- No need to answer the question stated in the introduction;
- If you start by developing your solution locally, it must be containerized;
- You need to provide a document/readme explaining your proposal, technical decisions, and constraings.
- You need to provide the diagram of your solution and the corresponding data model.
- There is no need for paid, expensive and highly performant data warehouses. You can use a "standard" SQL database.

Bonus points

- Provide IaC on Terraform or AWS CDK.

Delivery of your solution

The output of this exercise is expected to be a **private** GitHub repository and so, please add the following users to the repo:

- `henriqueribeiro`
- `peter-ram`
- `taxuspt`
- `alejogm0520`

Reviewing

There are no right or wrong answers to this challenge, the results you achieve are of secondary importance. We will mostly focus on: your work ethics, your skills, how you justify your design choices and your thought process. With this in mind, it's not mandatory that you provide a full fledged solution.

Below we leave some bullet points that we are looking for in the solution, so it can guide you through this challenge and help you to prepare for our discussion on your solution (we might not cover all but it should help):

- Data Engineering: Data formats, Ingestion techniques, Data Modelling, etc...
- Documentation: Is the project and the code properly documented?
- Technology: Which libraries or approaches are used? Do they make sense for the task? Justify why you've decided to use those technologies to solve the code challenge
- Code quality: Is the code understandable and maintainable?