

## Coding Challenge

At Grover we leverage Kafka as a bus for our microservices messages. As data engineers our job is to ingest data continuously in our data lake (based on S3) to make it available for different data functions. In this task we would like to leverage the power of faust (kafka streams for python) to ingest messages from Kafka in S3.

Please solve the challenge below by meeting the given acceptance criteria. When you have finished please ZIP your solution and do not publish your solution in any public GitHub repository or anywhere else!

In general, feel free to make assumptions and have fun with the challenge! Please make sure you focus your time on explaining and elaborating your thought process, assumptions, decisions, thoughts and comments, for example inside a `thought-process.md` file. Please also document how to build and run your code.

### Acceptance Criteria

- Add a faust Agent to consume users data and write to S3 in a format of your choice
- Implementation must be optimized to reduce the amount of S3 objects
- Data in S3 must be partitioned efficiently to reduce data scans from query engines

### Nice to have

- How it would be possible to read this written data using for example Athena

### Tech Stack

- Run the command `make up` to spin up the necessary components
  - producers: simple python producer that send messages continuously
  - consumers: faust consumers
  - kafka
- Use any libraries that can help you on the way
- Test code where you feel it makes sense
- Use Minio or Localstack to make API requests against S3, alternatively using a real S3 bucket is also fine