Instabug Backend Challenge

It's required to build a chat system. The system should allow creating new applications where each application will have a token(generated by the system) and a name(provided by the client). The token is the identifier that devices use to send chats to that application.

Each application can have many chats. a chat should have a number. Numbering of chats in each application starts from 1 and no 2 chats in the same application may have the same number. The number of the chat should be returned in the chat creation request. A chat contains messages and messages have numbers that start from 1 for each chat. The number of the message should also be returned in the message creation request. The client should never see the ID of any of the entities. The client identifies the application by its token and the chat by its number along with the application token.

Add an endpoint for searching through messages of a specific chat. It should be able to partially match messages' bodies. You must use ElasticSearch for this.

The applications table should contain a column called chats_count that contains the number of chats for this application. Similarly, the chats table should contain a column called messages_count that contains the number of messages in this chat. These columns don't have to be live. However, they shouldn't be lagging more than 1 hour.

Assume that the system is to receive many requests. It might be running on multiple servers in parallel and thus multiple requests may be processed concurrently. Make sure to handle race conditions. Try to minimize the queries and avoid writing directly to MySQL while serving the requests(especially for the chats and messages creation endpoints). You can use a queuing system to achieve that. It is allowed for chats and messages to take time to be persisted. You should optimize your tables by adding appropriate indices.

Your app should be containerized. We should only write `docker-compose up` to run the whole stack.

Notes:
- You're only asked to build the API. No GUI is required.
- You should use Ruby on Rails(V5) for building the API and the workers. BONUS: You are encouraged to have the endpoints of chats and messages creation as a Golang app.
- The endpoints should be RESTful. You should provide endpoints for creating, updating and reading applications, chats that belong to a specific application(GET /applications/[application_token]/chats) and messages that belong to a specific chat. Make sure to have appropriate indices in place for these endpoints to be optimized.
- Use MySQL as your main datastore. You're allowed to use any other component you need along with MySQL. You may want to check out REDIS.
- You should add a Readme containing instructions to run your code.
- BONUS: Write specs for your code.

In case you have any questions. Please send an email to yelbazzar@instabug.com and ahamayed@instabug.com. After you're done with the task, put your code in a Github repo (or multiple repos if needed), then send the link of the repo to the same thread of emails. In case you prefer not to put the code publicly, feel free to send an archive containing all your code.