Feathery Coding Challenge

In order to help us assess your infrastructure experience and skills as a developer, please complete the exercise outlined below. We're looking for demonstration of:

- Thoughtful technical design
- Clean, well-structured, and performant code

In case something is unclear or you have any questions regarding the task, please do not hesitate to contact us at **peter@feathery.io.**

Happy coding!

Scalable External API Infrastructure

In this exercise, you will set up scalable server-side infrastructure on AWS for handling heavyweight, outbound API requests. You'll have 24 hours to complete this exercise, and feel free to use your discretion in choosing any resources to include in your project.

You will need to create an AWS account if you don't have one already. We will reimburse you up to $10 for any AWS costs incurred during this project.

The exercise will progress as follows:

1. Set up a simple web server (in your language and framework of choice) that takes in a multi-page PDF file and uses AWS's Textract API to extract the text. The file and extracted text should be saved to the database.
2. This web server should be deployed to AWS via Elastic Beanstalk, backed by a RDS Aurora Postgres database.
3. Make sure to design your infrastructure to be:
    a. Scalable
    b. Secure
    c. Easy to maintain
4. The endpoint should run quickly, efficiently, and be easily maintainable as well.
5. Once you finish, please commit your code to a repository on GitHub and invite the following accounts as collaborators: **bo-dun-1**, **zackkhan**, **ondrosh**, **ericpias**
6. You'll conduct a walkthrough of your infrastructure setup, code, and application. You'll be expected to cover the following:
    a. An overview of your implementation and how it works. Be technical and specific regarding important aspects of your solution but don't get lost in details. Try to formulate your explanations in a way that addresses other developers who might use your code in their projects.
    b. Walk us through your design process. How did you design your infrastructure setup? How did you approach testing? Did you make any decisions on engineering tradeoffs? If so, please walk us through your decision making process.
    c. Try to identify and discuss limitations of your implementation. Can you think of a situation in which your implementation might suffer from degraded performance or behave in ways that a user would not expect? Are there edge cases or similar issues to be aware of?
    d. If you were not able to solve some parts of the task outlined below, please outline what is missing and try to identify why the particular part of the task is challenging.