Senior Data Engineer Technical Challenge

This challenge is a good representation of the type of work you will be doing within the role (except the data is already clean). The challenge is prescriptive where it needs to be so make sure you follow the instructions. If you get stuck or something is unclear, use your best judgement to work through the challenge. Minimum technology requirements

- Use the Python 3.x
- Make sure you use Pandas

Good luck!

The Challenge

1. Download the raw working file here
2. Write a script or notebook to perform the following operations on the data
   a) Extract the data from the CSV
   b) Normalize **MovementDateTime** to ISO format
   c) For each ship grouped by **CallSign**, if the **MoveStatus** is "Under way using engine", fill in any missing or zero speeds with the average of all speeds for that **CallSign**
   d) Create a new feature called **BeamRatio** calculated as Beam / Length (Beam divided by Length)
   e) Save all the existing features and newly calculated features in steps b, c & d as a new csv file (enriched.csv) on your local machine.
   f) Use the data file enriched.csv and store it in a Postgres or MySQL table on your local machine (**NOTE:** You are expected to write python code to programmatically insert the data into the database)
   g) Provide a screen shot of the data inserted into the table
3. Wrap all your code & enriched.csv into a private repo on Github and add **khordoo-m** and **cr3** as collaborators

Step A (required)

You're expected to setup an AWS account for this section. The free tier will cover all the requirements so you won't need to pay anything. If you're out of capacity in the free tier, the challenge should not cost more than a few dollars.

Minimum technology requirements

- Boto3 is your friend
- The AWS CLI is required to move the CSV from local to a S3 bucket

4. Install and configure the [AWS CLI] (https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html) for your environment
5. Write a script (bash preferred) that uploads the **pace-data.csv** file into an S3 bucket (bonus points for creating the bucket through the CLI). If you don't want to write a script, write down the commands you use to create the bucket and upload the file.

Step B (required)

6. Write a Lambda handler that gets triggered when a file that matches the filename (i.e **pace-data.csv)** lands in the bucket you created above.
7. This lambda function is responsible for executing the operations described in step 2. (wrap the code you wrote in step 2 to be executed within the lambda)

   The code should perform the following steps:

   - Operations 2a to 2d,
   - Write **enriched.csv** back into the s3 bucket

Bonus Point (not mandatory)

8. All of the above
9. Create an RDS database (Postgres or MySQL)
10. Extend the Lambda Function
    - Add the necessary code to insert the contents of enriched.csv into the RDS database.
    - Ensure the Lambda function both saves the file to S3 and inserts its contents into the database.

Double Bonus Points (not mandatory)

1. All of the above.
2. Use an Infrastructure-as-Code tool (e.g., CloudFormation) to create all AWS resources.
3. Set up a CI/CD pipeline (GitHub Actions, Azure DevOps Pipelines, etc.) that automatically deploys the resources whenever changes are pushed to the master/main branch.

Triple Bonus Points (not mandatory)

- All of the above
- Find and use a map API to track the course of ship with CallSign 5BUU3