Platform Engineer Coding Test

Intro

Time estimate

- The test is tailored for ~**55 minutes**, you are welcome to take more time. Taking extra time will not affect your evaluation.
- There is no deadline, however, we will review in batches, on a first-come-first-served basis.

Screen and audio recording

- To ensure a consistent and thorough evaluation process for all candidates, we request you to **screen share** and **record video/audio** while you work.
- Feel free to **use your favorite recording software**.
    - We recommend using a Zoom Personal Free account, which comes with screen sharing, audio recording, and saving to local. You should be able to complete the exercise within the duration of the meeting. However, if you need more time, you can save and share multiple recordings.
- By submitting your code and recordings, you consent to their use solely for evaluation purposes.

Handling questions and assumptions

- Due to the high volume of applicants, **we are unable to respond to any questions about the coding test**.
    - If you face any challenges, make assumptions and deviations as you see fit, include your explanation in README. You are welcome to use another stack or alternative to achieve a similar objective.

Tasks

Scenario

- You are working as a Platform Engineer at a company that has recently experienced intermittent issues with their web application's performance.
- Your task is to create a monitoring and automation solution that helps identify and resolve these issues automatically.

Submission Guidelines (10 minutes)

- **We recommend sharing a link to your solution in a GitHub repository.** Make sure the repository and links are publicly **accessible**.
    - If you do not have a GitHub account, you can share using any other method.
- Here is a recommended structure for your repository:
    - **Sumo Logic Query:** A text file (sumo_logic_query.txt) containing the Sumo Logic query.
    - **Lambda Function:** Python code (lambda_function/) for the AWS Lambda function.
    - **Terraform Configuration:** Terraform files (terraform/) used to deploy infrastructure resources.
    - **Screen and Audio Recordings:** Links to download the recordings.
    - **README File:** Brief documentation of the tasks.

Part 1: Sumo Logic Query and Alert (10 minutes)

- **Implement a Sumo Logic Query:**
    - Write a query to identify log entries where the response time of the '/api/data' endpoint exceeds 3 seconds.
    - Create an alert that triggers if more than 5 such entries are detected within a 10-minute window.
- **Screen and Audio Recording:**

- Record your screen while you implement the query and set up the alert.
- Narrate your thought process as you work.

Part 2: AWS Lambda Function (20 minutes)

- **Lambda Function Implementation:**
  - Write a Python AWS Lambda function that gets triggered by the Sumo Logic alert.
  - The function should restart a specified EC2 instance, log the action, and send a notification to an SNS topic.
- **Deployment and Testing:**
  - Deploy the Lambda function using the AWS Management Console or CLI.
  - Test the Lambda function to ensure it restarts the EC2 instance and sends the notification.
- **Screen and Audio Recording:**
  - Record your screen while you implement, deploy, and test the Lambda function.
  - Narrate your thought process as you work.

Part 3: IaC Setup (15 minutes)

- **Terraform Configuration:**
  - Write a Terraform script to deploy the EC2 instance, Lambda function, and SNS topic.
  - Optional/Bonus: Ensure that all resources are created with the least privilege necessary.
- **Deployment and Verification:**
  - Deploy the infrastructure using Terraform.
  - Verify that the resources are correctly created, and the Lambda function works as expected.
- **Screen and Audio Recording:**
  - Record your screen while you write, deploy, and verify the Terraform script.
  - Narrate your thought process as you work.

Thank you

We know your time is valuable, so thank you for your effort in showcasing your skills!