Interview Challenge

Dear candidate
Welcome to the Java Platform Group interview challenge.

Instructions

1) Please read through the entire challenge sheet before starting. Some of the parts of the challenges are inter-related, and may affect how you approach the problem.
2) You can use any means in your search for a solution. There are different solutions to solve the problem and you will be expected to justify your engineering choices.
3) As mentioned in the earlier email, you should have your Oracle Cloud Infrastructure Free account ready for this challenge.

Tip: Don't over-engineer your solution.

Prerequisites

- An oracle cloud account. We recommend that you sign up for a trial account with the free $300 credits (and the free tier resources). We are not able to assist you in creating / provisioning a new account, or if you have issues accessing the service.

- A GIT repository that you can store your solution and share the link with us.

Challenge 1: Coding

This challenge requires you to develop a REST endpoint for a survey management system. The backend will forward requests into OCI Logging Service so that data can be further analysed in OCI.

Requirements:

1. The endpoint will match the swagger specification outlined below.
2. The endpoint will be implemented using DROPWIZARD (see references below).
3. Upon reception of a request, the program will output the values to console.
4. The program will send each data point received to the OCI Logging Service (ingestion API) using the OCI JAVA SDK (See references below).
5. The program will build using maven.

This challenge is complete if you are able to demonstrate this running locally.

Challenge 2: Containerization

The solution must also be run from a docker container. Include the necessary files and information to build such a docker container.

This challenge is complete if you are able to demonstrate this running locally on a docker container.

Challenge 3: Testing

In order to test your endpoint and be able to visualize data in the cloud, you will need a way to generate and send random HTTP samples to your endpoint.

The goal of this challenge is to use any mean of your choice (programming language, program, framework, script, …) to build this sample generator.

Challenge 4: Search and visualize

In the OCI logging service, create a query that will display as a table the average score per gender.
Document the OCI CLI command(*logging-search*) that will execute the same query on a remote host.

In the OCI logging service, create a visualization that displays the number of surveys per gender.

Challenge 5: Documentation

In a README.md in the root of you repository, provide documentation of your solution so that other developers can understand, maintain, run, and deploy it on their own.

Swagger spec

```
paths:
    /surveys:
        post:
            summary: Send survey information
            requestBody:
                required: true
                content:
                    application/json:
                        schema:
                            $ref: '#/components/schemas/SurveyData'

components:
    schemas:
        SurveyData:
            type: object
            properties:
                age:
                    type: integer
                gender:
                    type: string
                    enum: [male, female, other]
                region:
                    type: string
                surveyID:
                    type: string
                score:
                    type: integer
                    minimum: 1
                    maximum: 5
            required:
                - age
                - gender
                - region
                - surveyID
                - score
```

References

1. Dropwizard (https://www.dropwizard.io/en/latest/index.html)
2. OCI SDK for java: https://docs.oracle.com/en-us/iaas/Content/API/SDKDocs/javasdk.htm#SDK_for_Java
3. OCI Logging ingestion API: https://docs.oracle.com/en-us/iaas/Content/Logging/Concepts/custom_logs.htm