CHECK POINT

DevOps candidate's Home Assignment

As part of your recruitment process, you are required to complete and submit the following DevOps home assignment

Introduction:

This assignment requires you to create a system of two Docker microservices (using Python or another language you are familiar with) deployed on ECS, utilizing an S3 bucket, an Elastic Load Balancer, and SQS.

You will also use GitHub\GitLab\Azure Devops for automation.

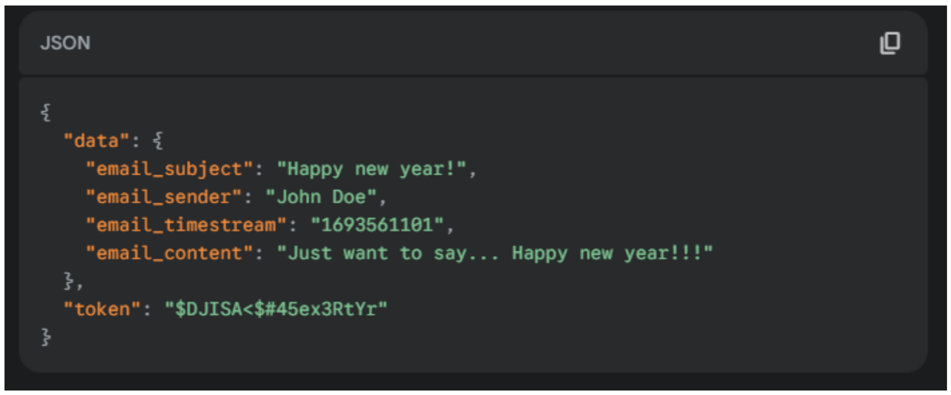
All infrastructure must be defined using Infrastructure as Code (IaC) with CloudFormation or Terraform.

Tasks:

1. Set up the chosen CI/CD tool to manage the CI/CD processes.
2. Provision your AWS cloud environment using an IaC tool (CloudFormation or Terraform).
3. Create an SQS queue and an S3 bucket.
4. Microservices creation:

   Microservice 1:

   - Create a REST microservice that receives requests from an ELB and forwards them to the backend service. The service should listen on a designated port. Example request payload:



```json
{
  "data": {
    "email_subject": "Happy new year!",
    "email_sender": "John Doe",
    "email_timestream": "1693561101",
    "email_content": "Just want to say... Happy new year!!!"
  },
  "token": "$DJISA<$#45ex3RtYr"
}
```

After token validation, publish the payload data to the SQS queue.

The token should be retrieved from SSM Parameter Store for comparison. The validation process should verify:

- Token correctness
- Date validity (ensure the email_timestream field is present and correctly formatted)

Microservice 2:

- Create an SQS microservice that periodically pulls messages from the SQS queue and uploads them to the designated path in the S3 bucket. Configure the pulling interval as needed.

5. **CI Jobs:** Create CI jobs for both microservices. Each job should build the respective Docker image and push it to a Docker repository (Docker Hub, ECR, etc.).

6. **CD Jobs:** Create CD jobs for both microservices. Each job should retrieve the image version and deploy it to the environment created using IaC.

Bonus:

1. Implement tests for the entire process.
2. Integrate monitoring tools (e.g., Grafana, Prometheus, or similar) to monitor the CI/CD process and the microservices' activity.

<u>Important Considerations:</u>

- Your work should be executable and clearly documented to allow for testing. Include a README file with instructions on how to run the code.
- All code must be pushed to a public Git repository. Provide the repository link when submitting the assignment.
- The README file should be comprehensive and easy to follow.
- All AWS resources can be created in your personal AWS account **using only free-tier instances**, or by using LocalStack.