PEARLTHOUGHTS IaC Task

Design the IaC (Terraform, Aws ECS/Fargate) for Medusa open source headless commerce platform backend (https://docs.medusajs.com/deployments/server/general-guide), CD pipeline using GitHub Actions.

To implement Infrastructure as Code (IaC) for the Medusa open-source headless commerce backend using Terraform and deploy it on AWS ECS/Fargate, follow these steps:

Terraform Setup:

1.**Create a VPC**: Define a Virtual Private Cloud (VPC) with subnets, internet gateway, and route tables.

2.**Create ECS Cluster**: Define an ECS cluster and configure Fargate as the launch type.

3.**Task Definition**: Create an ECS task definition with Docker container specifications for Medusa.

4.**ECS Service**: Set up an ECS service that runs the task definition, with load balancing and auto-scaling as needed.

5.**Security Groups:** Define security groups for your ECS service, allowing necessary inbound and outbound traffic.

6.**IAM Roles**: Define IAM roles with permissions for ECS tasks, including access to ECR, S3, etc.

GitHub Actions CI/CD Pipeline:

1.**Trigger**: Define triggers for the pipeline, such as pushing to the main branch.

2.**Checkout Code**: Use actions/checkout@v2 to pull the latest code from the repository.

3.**Setup Terraform**: Install Terraform and initialize it using terraform init.

4.**Build & Push Docker Image**: Build Medusa Docker image and push it to an AWS ECR repository.

5.**Deploy with Terraform**: Apply the Terraform configuration to deploy the infrastructure on AWS.

6.**Update ECS Service**: Use Terraform or AWS CLI to update the ECS service with the latest Docker image.

GitHub Actions Workflow:

```
name: Deploy to ECS Fargate

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v2

    - name: Setup Terraform
      uses: hashicorp/setup-terraform@v1

    - name: Terraform Init
      run: terraform init

    - name: Terraform Plan
      run: terraform plan

    - name: Terraform Apply
      run: terraform apply -auto-approve

    - name: Login to ECR
      run: aws ecr get-login-password | docker login --username AWS --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

- name: Build Docker Image

  run: docker build -t <image_name> .

- name: Push Docker Image to ECR

  run: docker push <aws_account_id>.dkr.ecr.<region>.amazonaws.com/<image_name>

- name: Update ECS Service

  run: aws ecs update-service --cluster <cluster_name> --service <service_name> --force-new-deployment.

THANK YOU