**How to set up carbon emissions tracking service on AWS**

**Step 1: Create an ECR repository**

aws ecr create-repository --repository-name energy-metrics-service

**Step 2: Create a new aws.yml for GitHub Actions deployment**

name: Deploy to Amazon ECS  
  
on:  
 push:  
 branches: [ "main" ]  
  
env:  
 AWS\_REGION: us-east-1  
 ECR\_REPOSITORY: energy-metrics-service  
 ECS\_SERVICE: energy-metrics-service  
 ECS\_CLUSTER: energy-metrics-cluster  
 CONTAINER\_NAME: energy-metrics-container  
  
permissions:  
 contents: read  
  
jobs:  
 deploy:  
 name: Deploy  
 runs-on: ubuntu-latest  
 environment: production  
  
 steps:  
 - name: Checkout  
 uses: actions/checkout@v3  
  
 - name: Configure AWS credentials  
 uses: aws-actions/configure-aws-credentials@v1  
 with:  
 aws-access-key-id: ${{ secrets.AWS\_ACCESS\_KEY\_ID }}  
 aws-secret-access-key: ${{ secrets.AWS\_SECRET\_ACCESS\_KEY }}  
 aws-region: ${{ env.AWS\_REGION }}  
  
 - name: Login to Amazon ECR  
 id: login-ecr  
 uses: aws-actions/amazon-ecr-login@v1  
  
 - name: Build, tag, and push image to Amazon ECR  
 id: build-image  
 env:  
 ECR\_REGISTRY: ${{ steps.login-ecr.outputs.registry }}  
 IMAGE\_TAG: ${{ github.sha }}  
 run: |  
 docker build -t $ECR\_REGISTRY/$ECR\_REPOSITORY:$IMAGE\_TAG .  
 docker push $ECR\_REGISTRY/$ECR\_REPOSITORY:$IMAGE\_TAG  
 echo "image=$ECR\_REGISTRY/$ECR\_REPOSITORY:$IMAGE\_TAG" >> $GITHUB\_OUTPUT  
  
 - name: Fill in the new image ID in the Amazon ECS task definition  
 id: task-def  
 uses: aws-actions/amazon-ecs-render-task-definition@v1  
 with:  
 task-definition: task-definition.json  
 container-name: ${{ env.CONTAINER\_NAME }}  
 image: ${{ steps.build-image.outputs.image }}  
  
 - name: Deploy Amazon ECS task definition  
 uses: aws-actions/amazon-ecs-deploy-task-definition@v1  
 with:  
 task-definition: ${{ steps.task-def.outputs.task-definition }}  
 service: ${{ env.ECS\_SERVICE }}  
 cluster: ${{ env.ECS\_CLUSTER }}  
 wait-for-service-stability: true

**Step 3: Create a task definition file task\_definition.json**

{  
 "family": "energy-metrics-service",  
 "networkMode": "awsvpc",  
 "requiresCompatibilities": ["FARGATE"],  
 "cpu": "256",  
 "memory": "512",  
 "executionRoleArn": "arn:aws:iam::[[SAMSUNG-SRIB-OFFICIAL\_ACCOUNT\_ARN]]:role/ecsTaskExecutionRole",  
 "taskRoleArn": "arn:aws:iam::[[SAMSUNG-SRIB-OFFICIAL\_ACCOUNT\_ID]]:role/ecsTaskRole",  
 "containerDefinitions": [  
 {  
 "name": "energy-metrics-container",  
 "image": "SAMSUNG\_SRIB\_GCIFS\_ECR\_REPO\_URI:latest",  
 "essential": true,  
 "portMappings": [  
 {  
 "containerPort": 8000,  
 "protocol": "tcp"  
 }  
 ],  
 "environment": [  
 {  
 "name": "EIA\_API\_KEY",  
 "value": "SAMSUNG\_SRIB\_API\_KEY"  
 },  
 {  
 "name": "POLL\_INTERVAL",  
 "value": "3600"  
 }  
 ],  
 "logConfiguration": {  
 "logDriver": "awslogs",  
 "options": {  
 "awslogs-group": "/ecs/energy-metrics",  
 "awslogs-region": "us-east-1",  
 "awslogs-stream-prefix": "ecs"  
 }  
 }  
 }  
 ]  
}

**Step 4: Create the necessary AWS resources using Terraform**

provider "aws" {  
 region = "us-east-1"  
}  
  
resource "aws\_ecs\_cluster" "main" {  
 name = "energy-metrics-cluster"  
}  
  
resource "aws\_ecs\_task\_definition" "app" {  
 family = "energy-metrics-service"  
 network\_mode = "awsvpc"  
 requires\_compatibilities = ["FARGATE"]  
 cpu = 256  
 memory = 512  
 execution\_role\_arn = aws\_iam\_role.ecs\_task\_execution\_role.arn  
 task\_role\_arn = aws\_iam\_role.ecs\_task\_role.arn  
 container\_definitions = file("task-definition.json")  
}  
  
resource "aws\_ecs\_service" "main" {  
 name = "energy-metrics-service"  
 cluster = aws\_ecs\_cluster.main.id  
 task\_definition = aws\_ecs\_task\_definition.app.arn  
 desired\_count = 1  
 launch\_type = "FARGATE"  
  
 network\_configuration {  
 subnets = [aws\_subnet.main.id]  
 security\_groups = [aws\_security\_group.ecs\_tasks.id]  
 assign\_public\_ip = true  
 }  
}  
  
# VPC and networking resources  
resource "aws\_vpc" "main" {  
 cidr\_block = "10.0.0.0/16"  
   
 enable\_dns\_hostnames = true  
 enable\_dns\_support = true  
}  
  
resource "aws\_subnet" "main" {  
 vpc\_id = aws\_vpc.main.id  
 cidr\_block = "10.0.1.0/24"  
}  
  
resource "aws\_security\_group" "ecs\_tasks" {  
 name = "energy-metrics-tasks-security-group"  
 vpc\_id = aws\_vpc.main.id  
  
 ingress {  
 protocol = "tcp"  
 from\_port = 8000  
 to\_port = 8000  
 cidr\_blocks = ["0.0.0.0/0"]  
 }  
  
 egress {  
 protocol = "-1"  
 from\_port = 0  
 to\_port = 0  
 cidr\_blocks = ["0.0.0.0/0"]  
 }  
}

**Step 5: Deployment Steps:**

terraform init  
terraform plan  
terraform apply

**Please ensure:**

* Push the code to GitHub and ensure these secrets are set in your repository:
* AWS\_ACCESS\_KEY\_ID
* AWS\_SECRET\_ACCESS\_KEY
* EIA\_API\_KEY=Z2nXgkONpQlcrXquu4yLBSKlbVzzalTwo6jmbmBI