cd Downloads/Devops

git init

git remote add origin <https://github.com/PremAI19/secure-microservice-app.git>

git add .

git commit -m "micro-service project with full project stucture"

git config --global user.email [mpremkumar19@gmail.com](mailto:mpremkumar19@gmail.com)

git config --global user.name "prem1903"

git commit -m "micro-service project with full project stucture"

git add .

git push origin main

git push origin main To https://github.com/PremAI19/secure-microservice-app.git ! [rejected] main -> main (fetch first) error: failed to push some refs to 'https://github.com/PremAI19/secure-microservice-app.git' hint: Updates were rejected because the remote contains work that you do not hint: have locally. This is usually caused by another repository pushing to hint: the same ref. If you want to integrate the remote changes, use hint: 'git pull' before pushing again. hint: See the 'Note about fast-forwards' in 'git push --help' for details.

git pull origin main --allow-unrelated-histories

git add .

git commit -m "Resolved merge conflicts between local and remote"



**Set Up GitHub OIDC in AWS**

1. Go to the **AWS Console → IAM → Identity Providers**.
2. Click **“Add provider”**.
3. Choose:
   * **Provider Type**: OpenID Connect
   * **Provider URL**: https://token.actions.githubusercontent.com
   * **Audience**: sts.amazonaws.com
4. Click **Next → Add provider**.

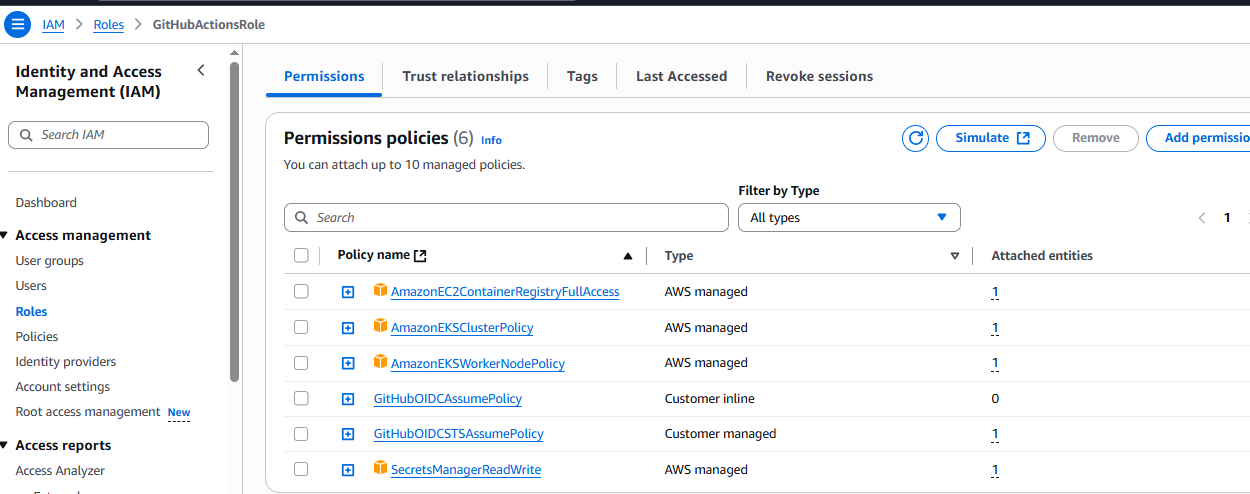
**Step 2: Create IAM Role for GitHub Actions**

1. Go to **IAM → Roles → Create role**
2. Select **Web identity** as trusted entity type.
3. Choose:
   * **Identity provider**: token.actions.githubusercontent.com
   * **Audience**: sts.amazonaws.com
4. In the “Conditions” section:
   * Click “Add Condition”
   * Select **StringEquals**:
     + token.actions.githubusercontent.com:sub
     + value: repo:YourGitHubUser/secure-microservice-app:ref:refs/heads/main
       - Replace with your **actual GitHub username and repo name**

✅ Example:

repo:PremAI19/secure-microservice-app:ref:refs/heads/main

1. Click **Next**
2. Attach policies:
   * AmazonEKSClusterPolicy
   * AmazonEKSWorkerNodePolicy
   * AmazonEC2ContainerRegistryFullAccess
   * SecretsManagerReadWrite
   * You can also create a custom policy for least-privilege.
3. Name the role: GitHubActionsRole
4. Finish and copy the **Role ARN**



**🔧 Step 3: Update Your GitHub Actions Workflow**

Replace the dummy ARN in .github/workflows/cicd.yml with the correct Role ARN:

yaml

- name: Configure AWS Credentials

uses: aws-actions/configure-aws-credentials@v4

with:

role-to-assume: arn:aws:iam::<your-account-id>:role/GitHubActionsRole

aws-region: us-east-1

Docker Build error

ERROR: unable to prepare context: path "./src/product-api" not found

means that the GitHub Actions runner cannot find the path ./src/product-api in your repo.

Dockerfile for the product service is inside product-service/, then you need to fix your GitHub Actions workflow:

- name: Build Docker image

run: |

docker build -t product-api:${{ env.IMAGE\_TAG }} ./product-service

docker tag product-api:${{ env.IMAGE\_TAG }} 835990278455.dkr.ecr.us-east-1.amazonaws.com/product-api:${{ env.IMAGE\_TAG }}

Docker scan

