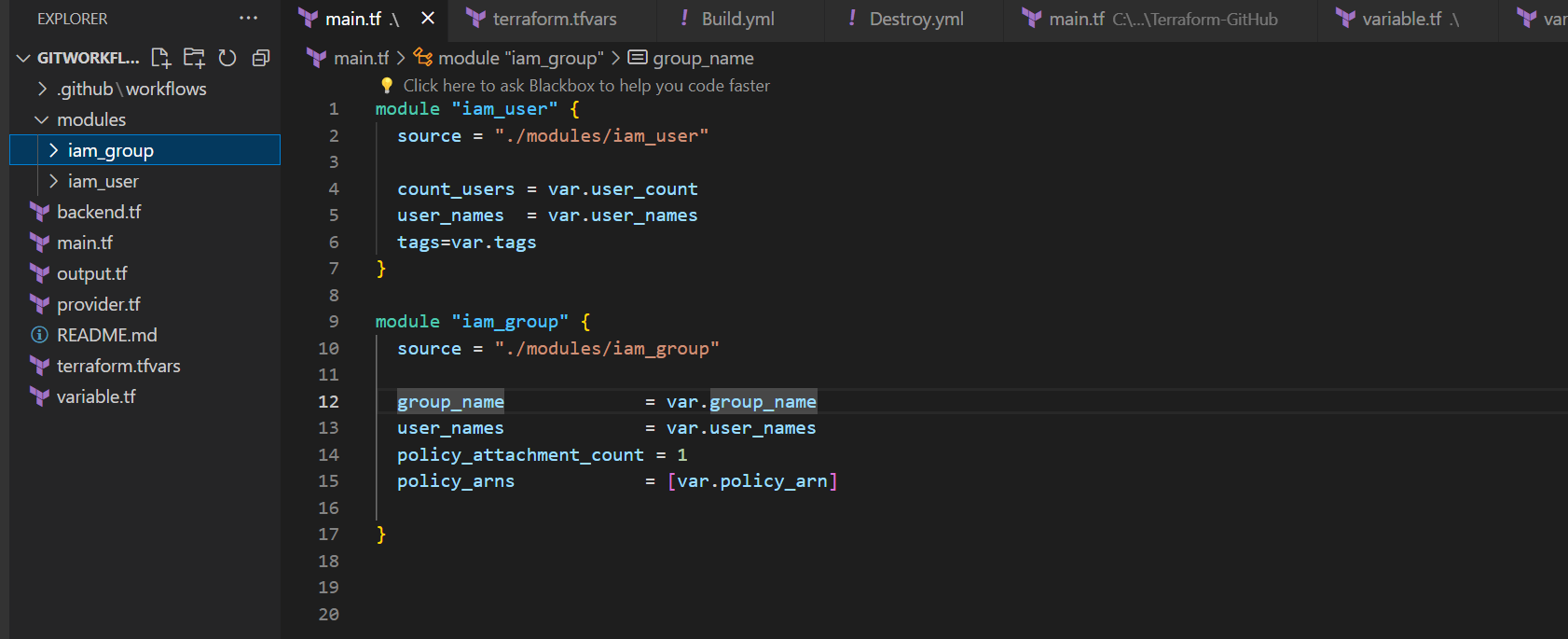
**DOCUMENTATION**

CREATE IAM USERS AND GROUP USING TERRAFORM AND DEPLOY IT THROUGH GITHUB WORKFLOW.

GITHUB LINK- [CLICK HERE!](https://github.com/Simi28/GitWorkflow-Assignment)

IAM users and groups are used together to manage access to AWS resources. By creating IAM users, assigning them to groups, and applying appropriate permissions, you can control who can access which AWS resources and what actions they can perform.

We Have Created IAM Users and Group in terraform using modular approach in Terraform.



And also we have added Backend.tf file to store .tfstate file in Remote location

So, to run this through GitHub workflow we need to Create a Workflow and over here we have create 2 workflows

1. Build.yml = It will Initialise ,Plan and Apply the resource

2.Destroy.yml =It will Destroy the Resources Created by Build

**Workflow for Build.yml-**

name: Workflow to Deploy Resource

on:

  workflow\_dispatch:

    branches: [ main ]

jobs:

  terraform:

    name: 'Terraform'

    runs-on: ubuntu-latest

    steps:

    - name: Checkout code

      uses: actions/checkout@v2

    - name: Set up 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: us-east-1

    - name: Setup Terraform

      uses: hashicorp/setup-terraform@v1

      with:

        terraform\_version: 0.13.4

    - name: Initialize terraform resources

      run: terraform init

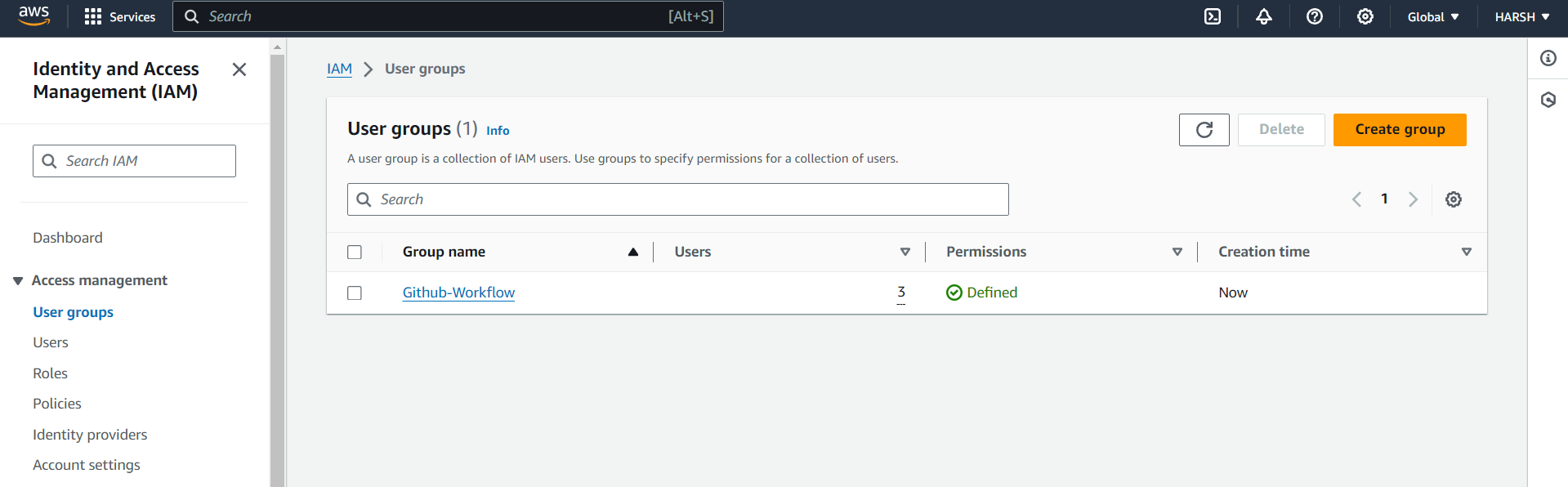
    - name: Plan the resource

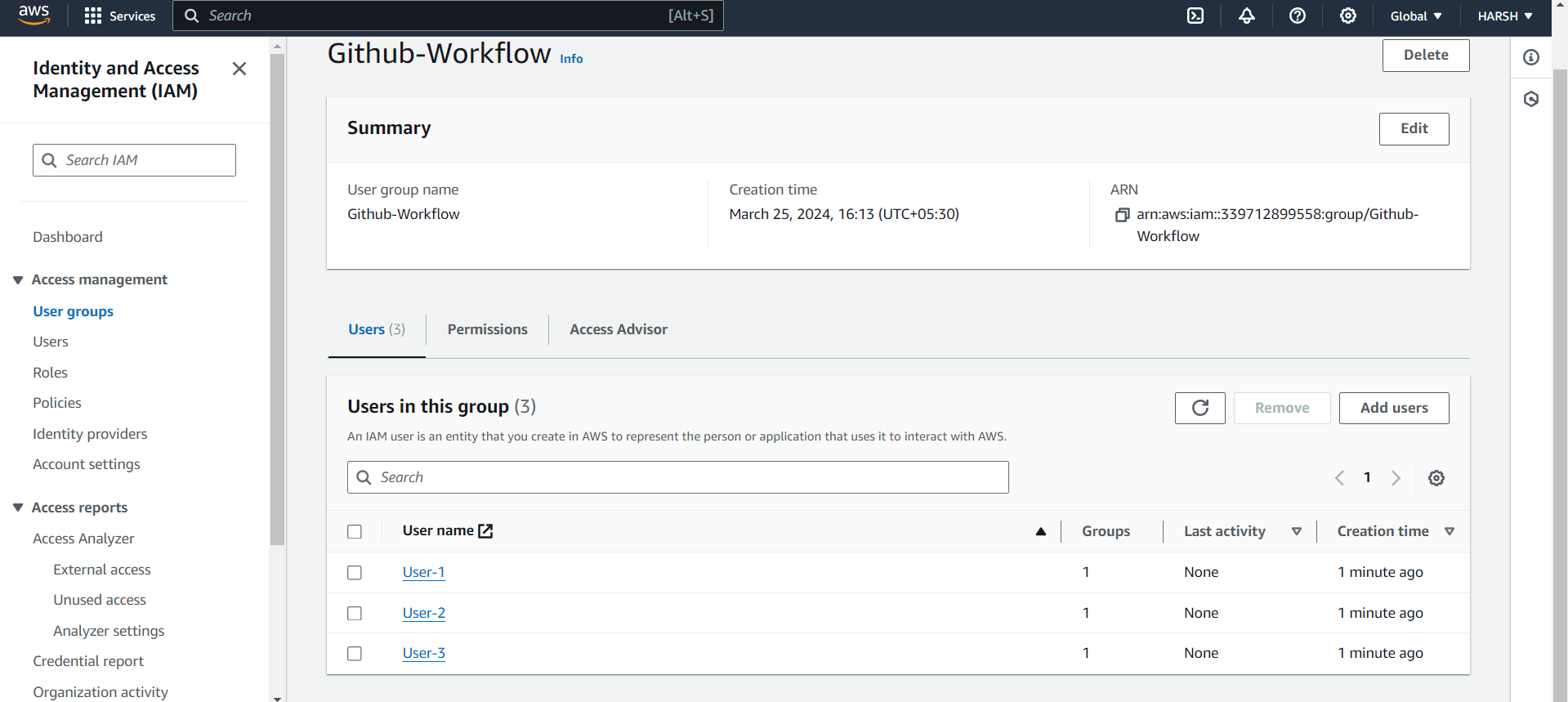
      run: terraform plan

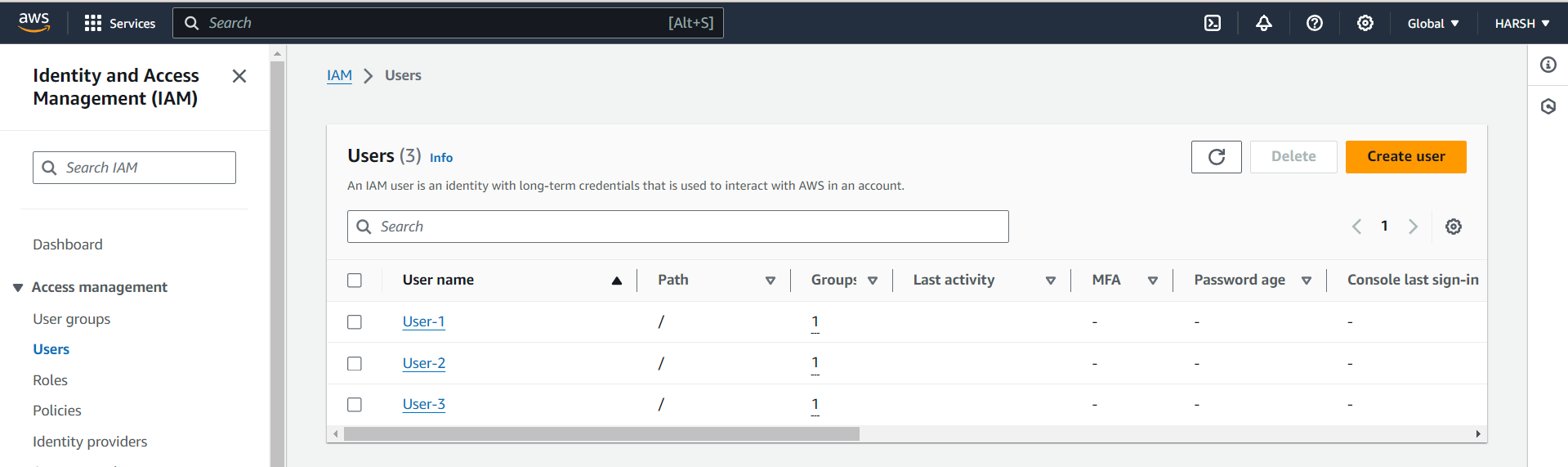
    - name: Apply the resources

      run: terraform apply -auto-approve

**OUPUT ON AWS CONSOLE-**

****

****

****

**WorkFlow for Destroy.yml-**

name: Terraform Destroy

on:

  workflow\_dispatch:

    branches: [ main ]

jobs:

  terraform:

    name: 'Terraform'

    runs-on: ubuntu-latest

    steps:

    - name: Checkout code

      uses: actions/checkout@v2

    # This step saves your AWS access and secret key to be used by Terraform

    - name: Set up 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: us-east-1

    - name: Setup Terraform

      uses: hashicorp/setup-terraform@v1

      with:

        terraform\_version: 0.13.4

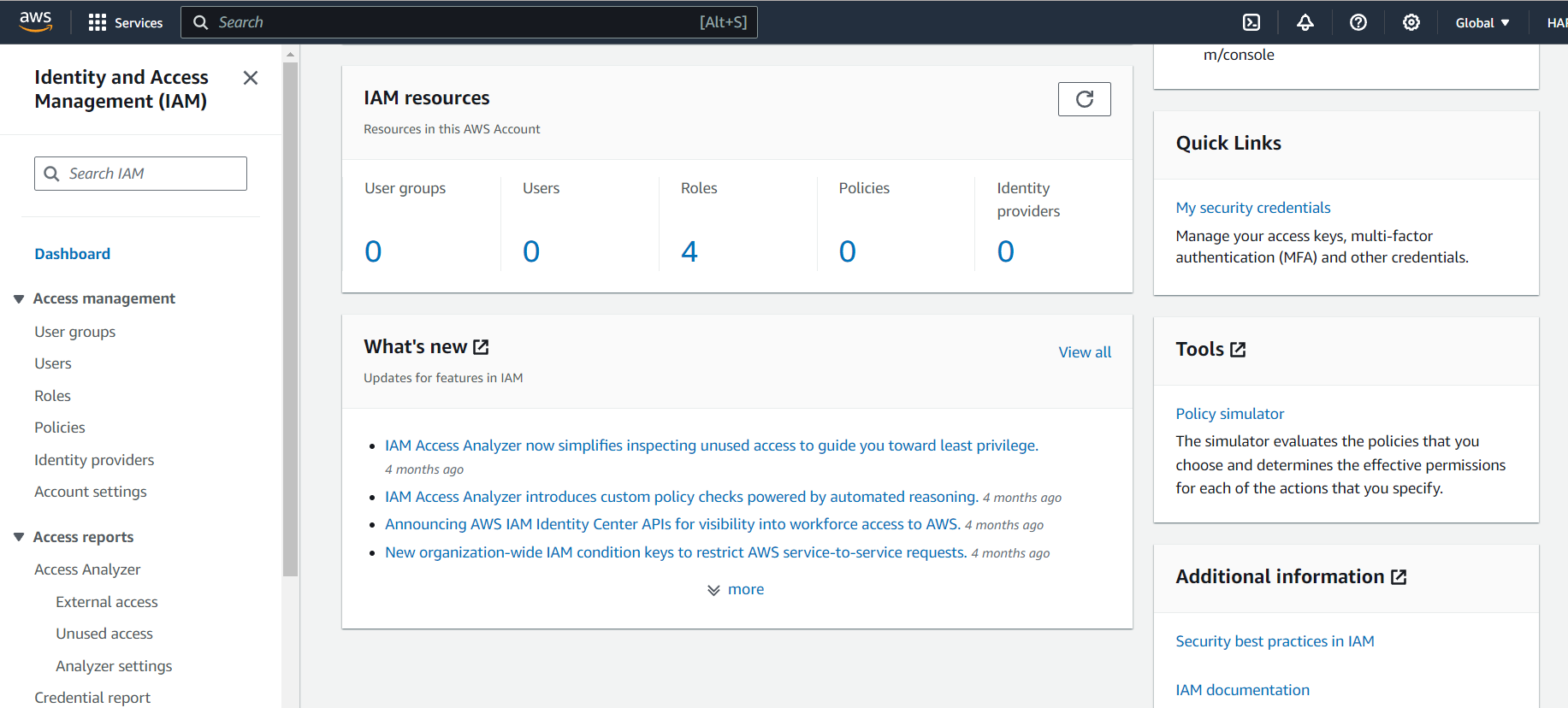
    - name: Initialize terraform resources

      run: terraform init

    - name: Destroy  Resources

      run: terraform destroy -auto-approve

**OUPUT ON AWS CONSOLE-**

****

**These both workflows are on workflow\_dispatch runner which allow you to run the workflow manually.**

**In this workflow we have used secrets so that we can store our Credential securely without exposing them in the code.**