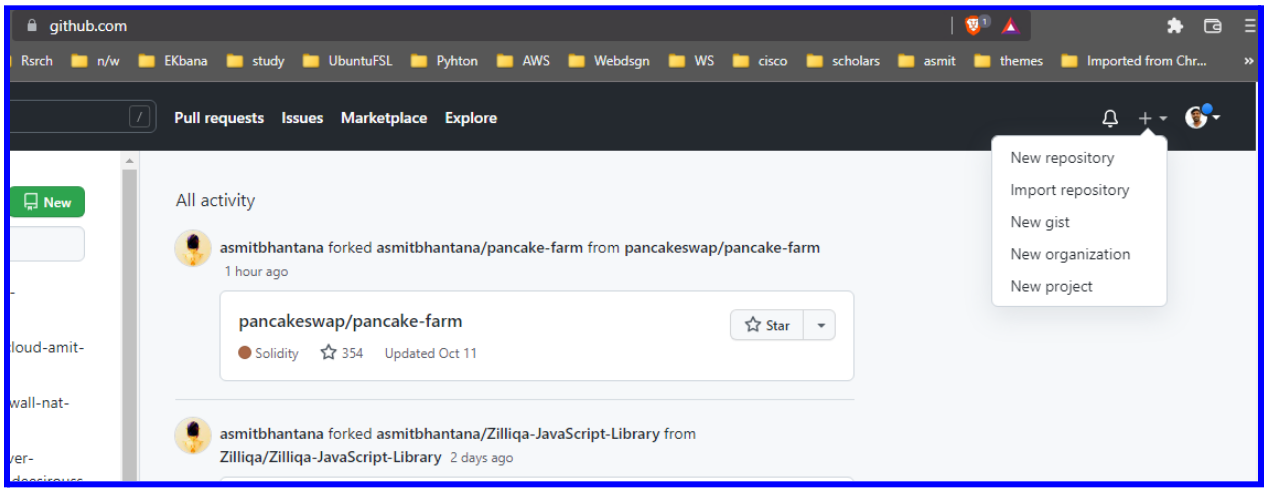
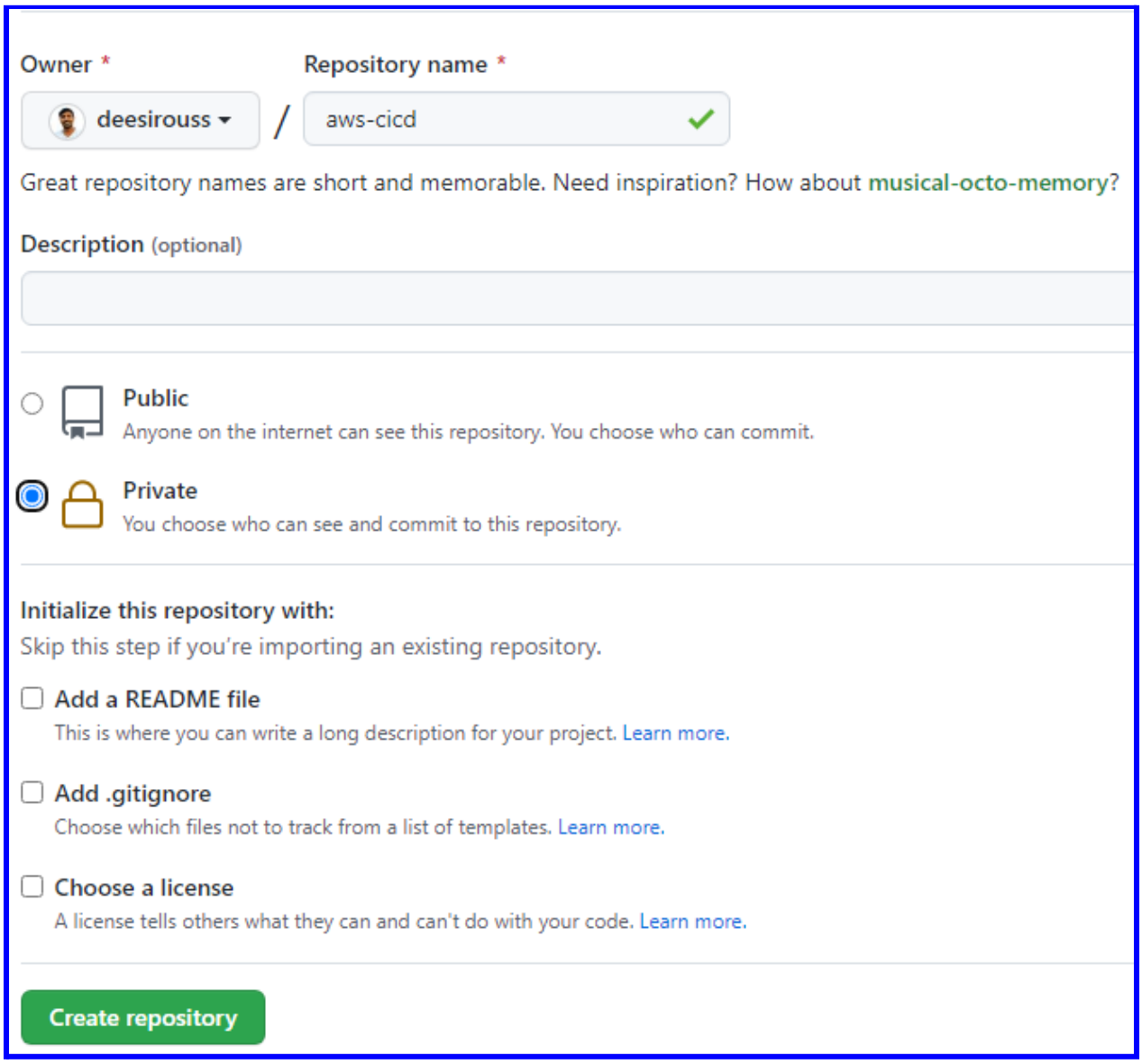
Integrate both these scripts with one of Jenkins, Github Actions, CircleCI or TravisCI

I have used **GitHub Actions**

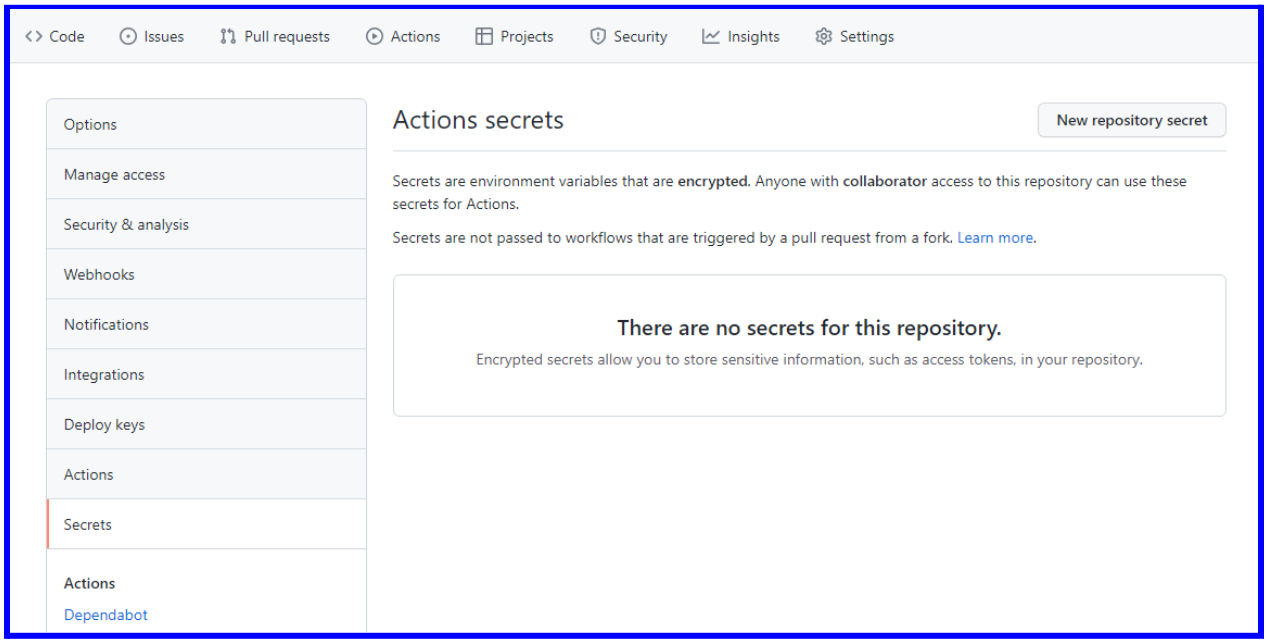
Creating new repository for Lambda CI/CD

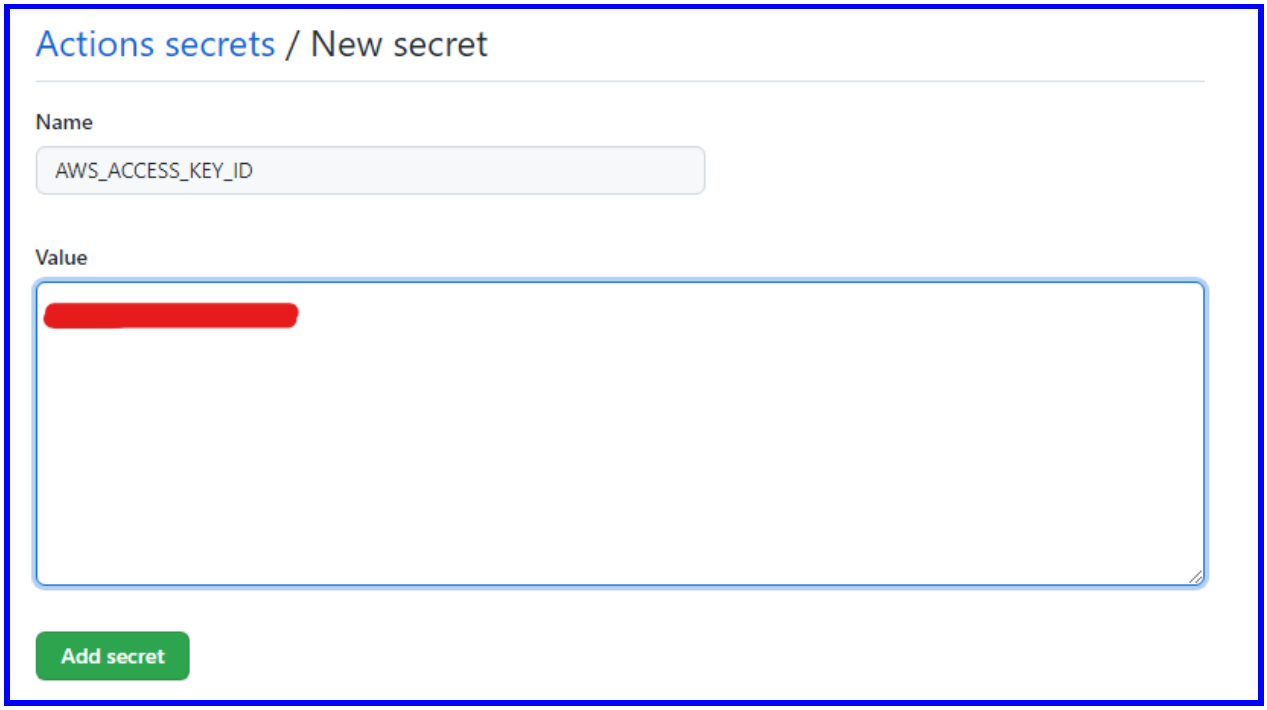


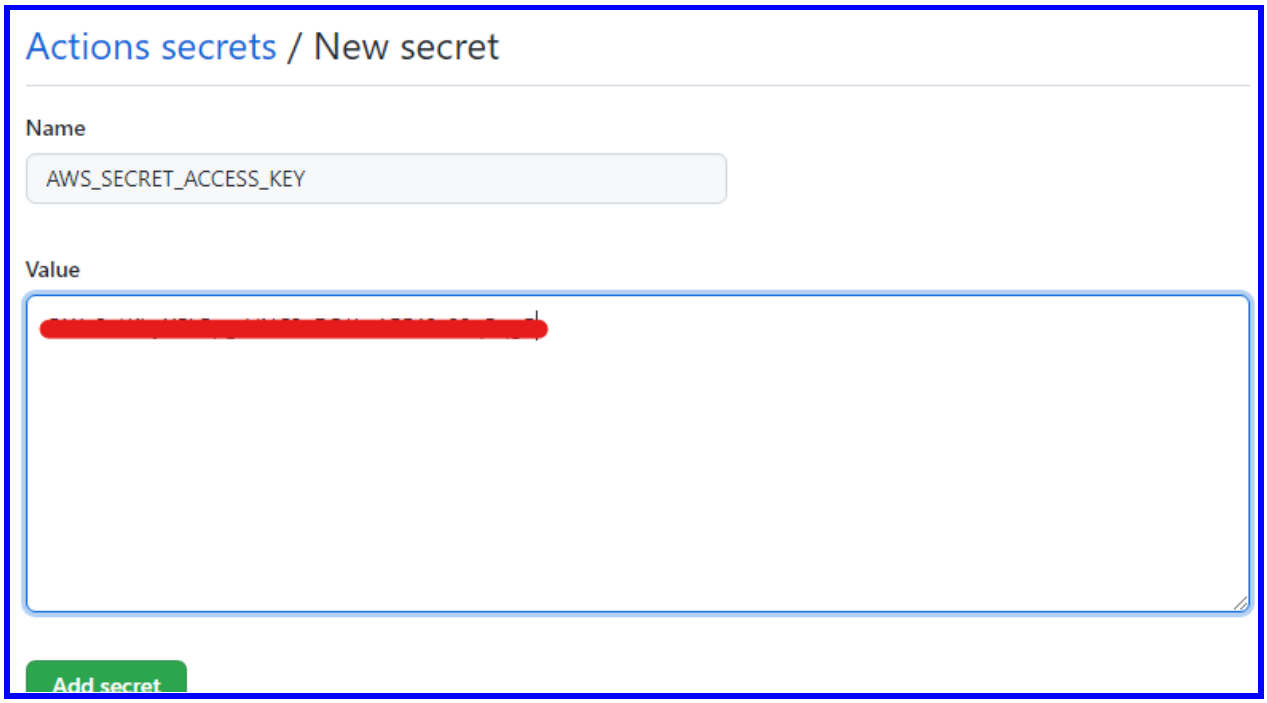
Name - aws-cicd, Private Repo



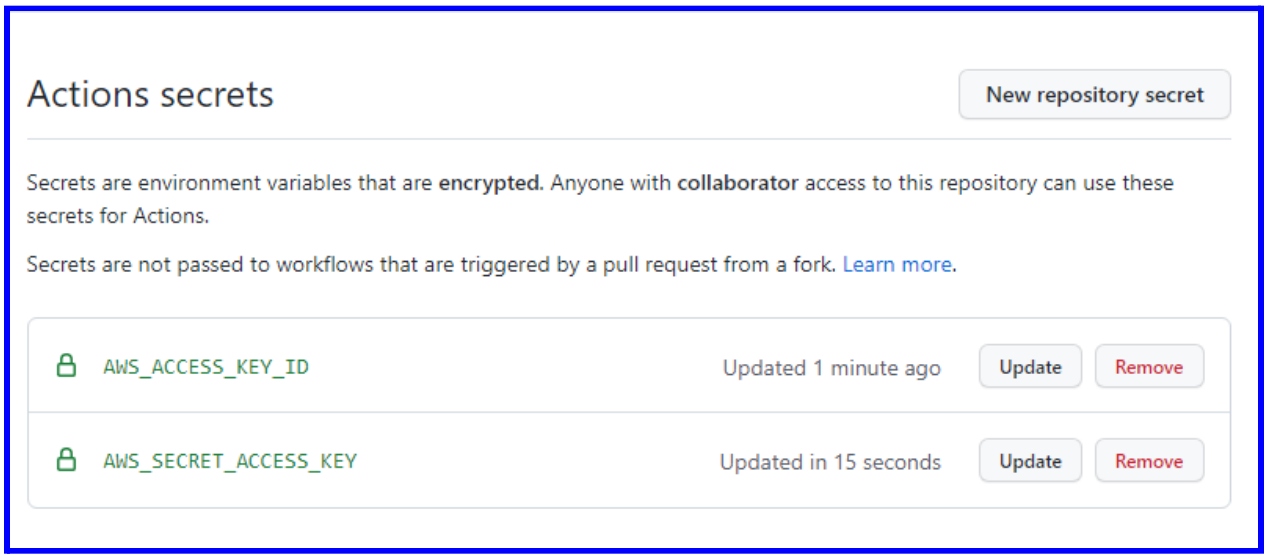
Creating Secret key for AWS Access key and Secret key







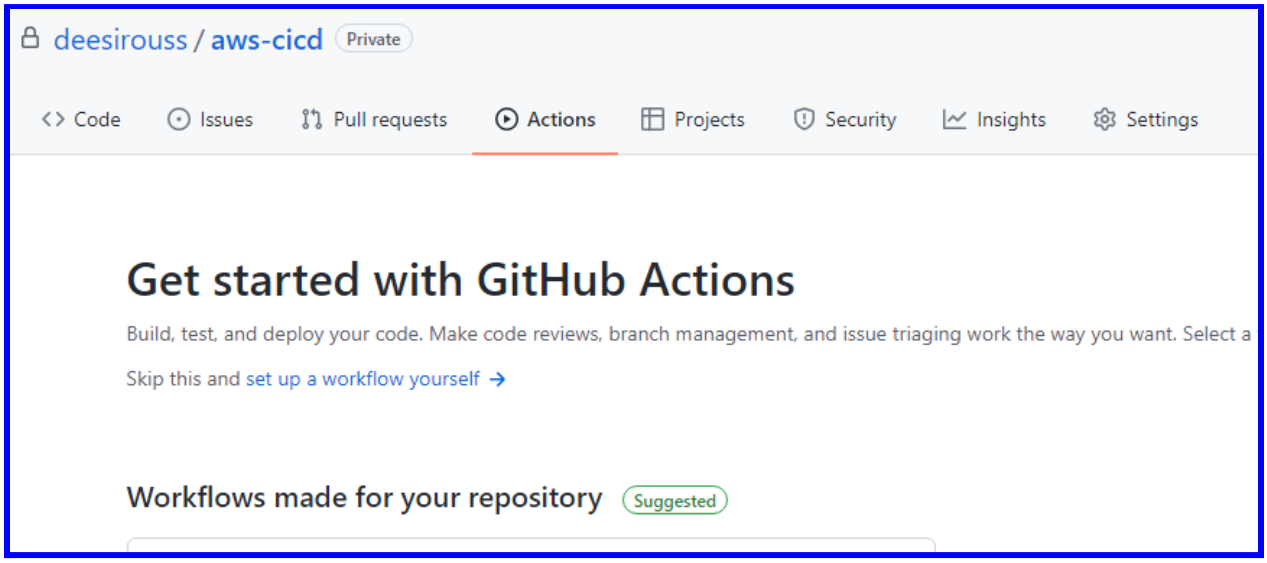
We have two Secrets



Added lambda_function.py in the github repository

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps({"Hello": "Default from Bibek Mishra"})
    }
```

Creating Github Actions → Actions → Set up a Workflow yourself

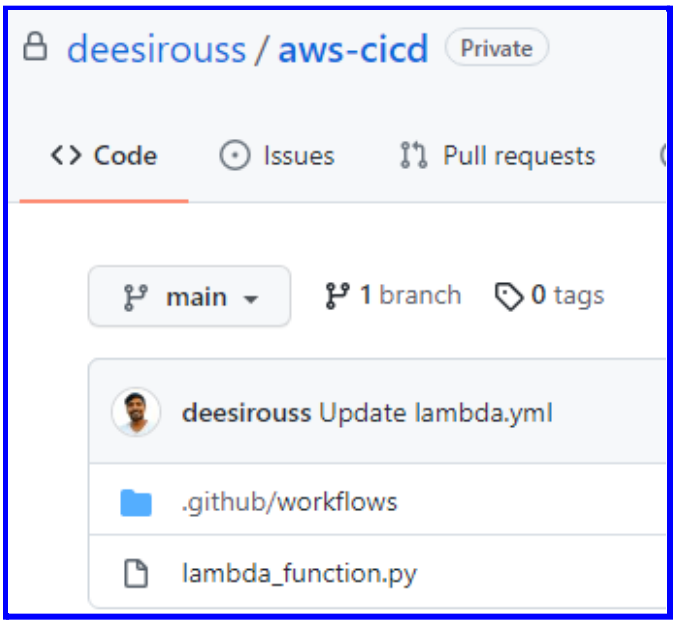


Workflow Yaml file

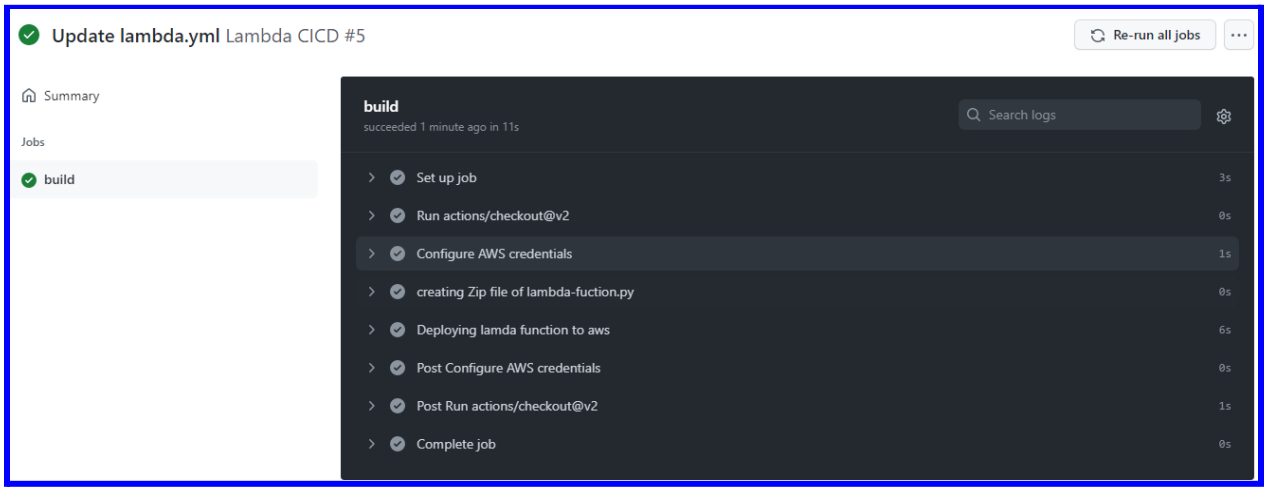
```
# This is a basic workflow to help you get started with Actions

name: Lambda CICD

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
```

```
# This workflow contains a single job called "build"
build:
  # The type of runner that the job will run on
  runs-on: ubuntu-latest

  # Steps represent a sequence of tasks that will be executed as part of the job
  steps:
    # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
    - uses: actions/checkout@v2

    #AWS credentials
    - 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 }}
        # TODO Change your AWS region here!
        aws-region: us-east-2

    - name: creating Zip file of lambda-fuction.py
      run: |
       zip lambda_function.zip lambda_function.py

    - name: Deploying lambda function to aws
      run: |
       aws lambda create-function --function-name bibek-demo-deployment \
       --runtime python3.9 --zip-file fileb://lambda_function.zip \
       --role arn:aws:iam::949263681218:role/service-role/bibek-api-gw-default-role-mof54wmn \
       --handler lambda_function.lambda_handler
```

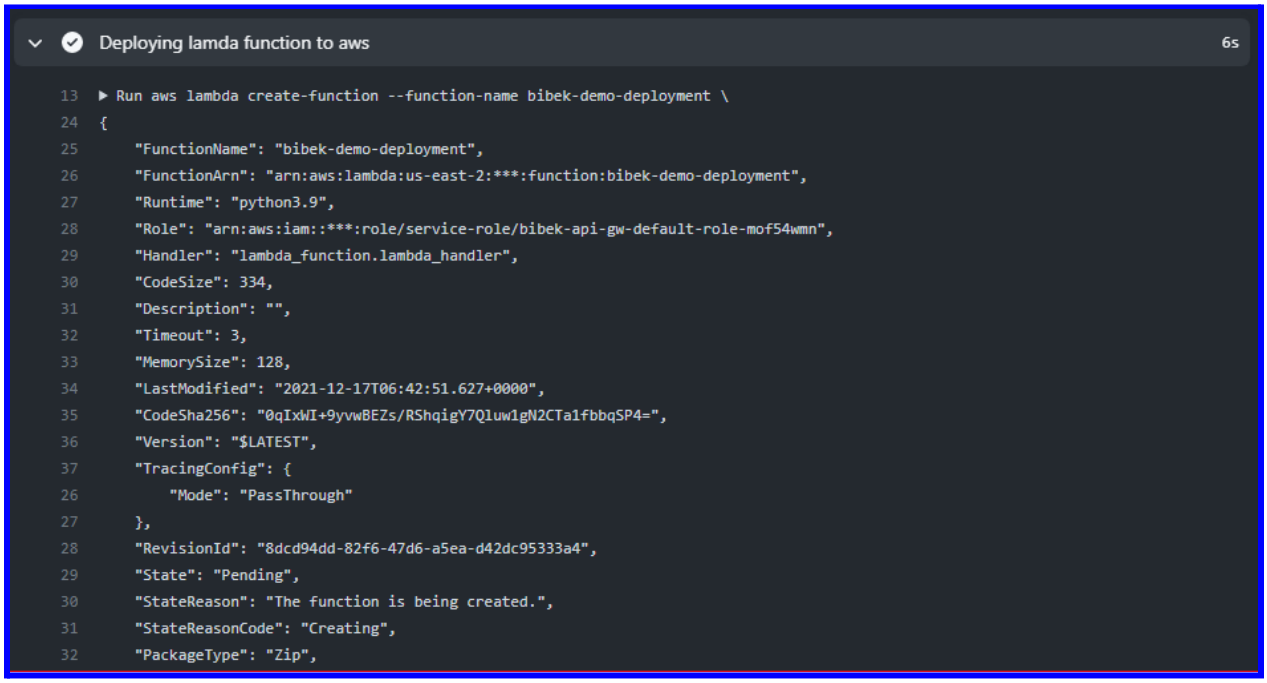
We have two files in our repository



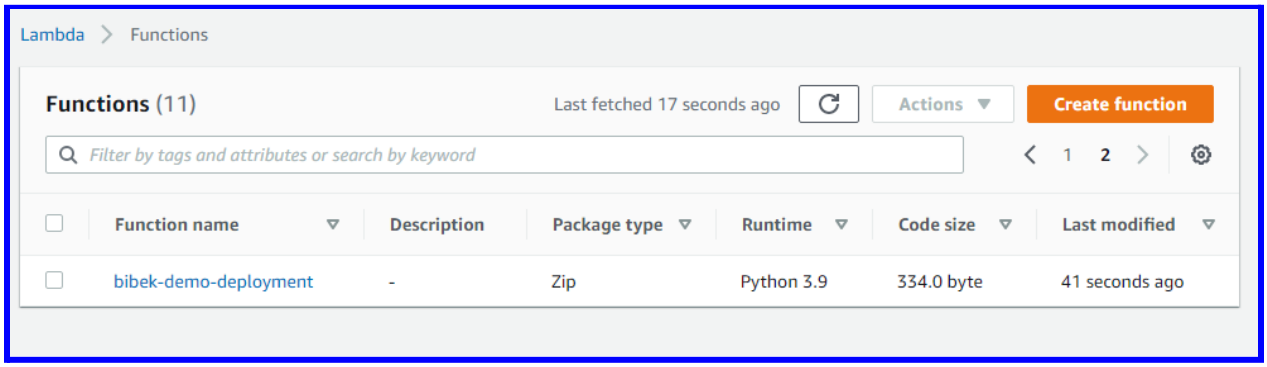
If we see the build result, all jobs have been executed successfully

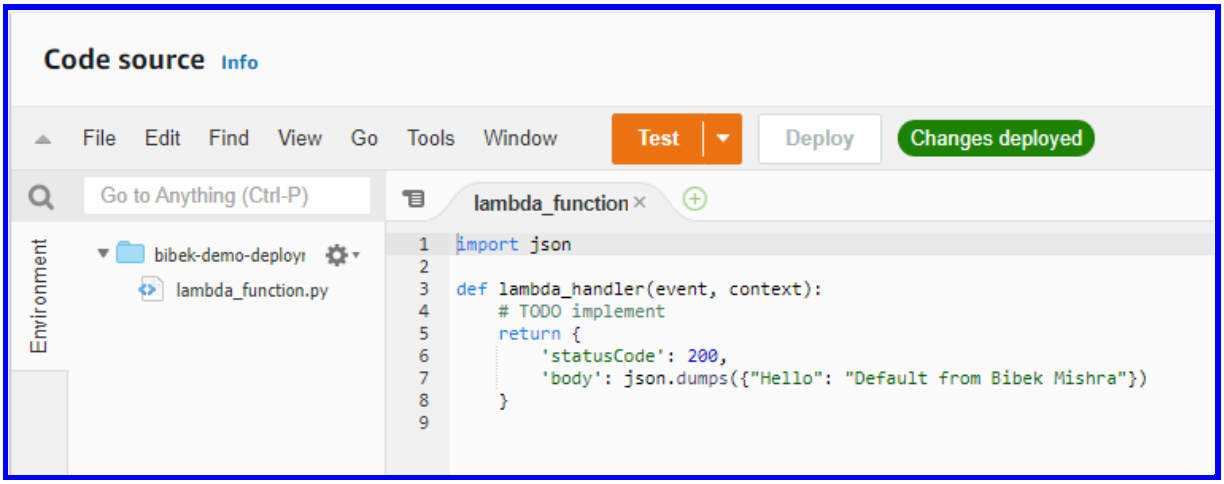


Deploying Lambda function to aws Job details



We can see the new lambda function has been deployed recently





Slight changes in json file for message

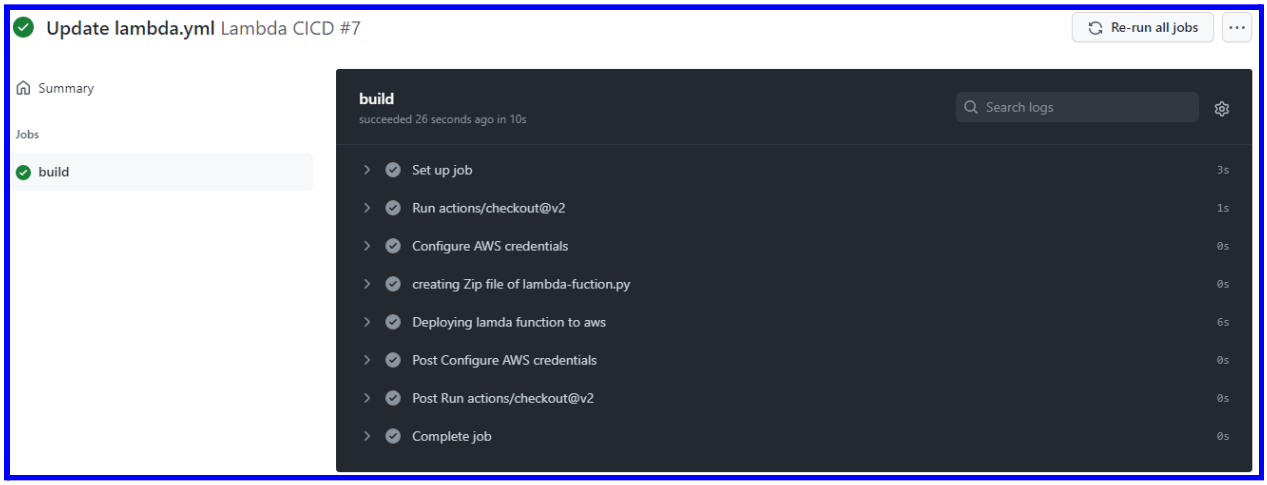
```
import json
def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps({"Hello": "CI/CD from Bibek Mishra"})
    }
```

Changing the workflow to update the created function

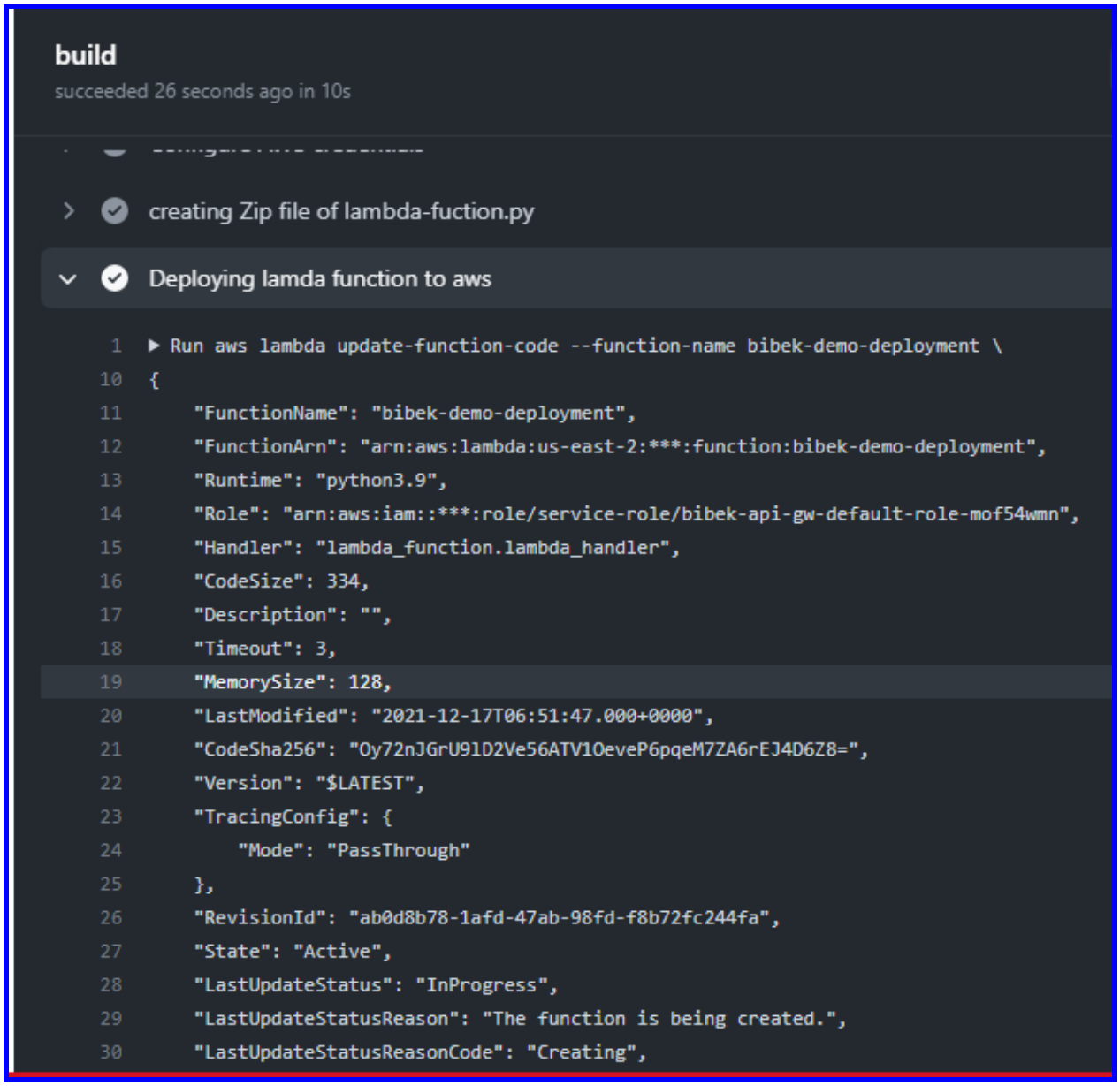
```
#AWS credentials
      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          # TODO Change your AWS region here!
```

```
      aws-region: us-east-2

    - name: creating Zip file of lambda-fuction.py
      run: |
       zip lambda_function.zip lambda_function.py

    - name: Deploying lambda function to aws
      run: |
       aws lambda update-function-code --function-name bibek-demo-deployment \
       --zip-file fileb://lambda_function.zip
```

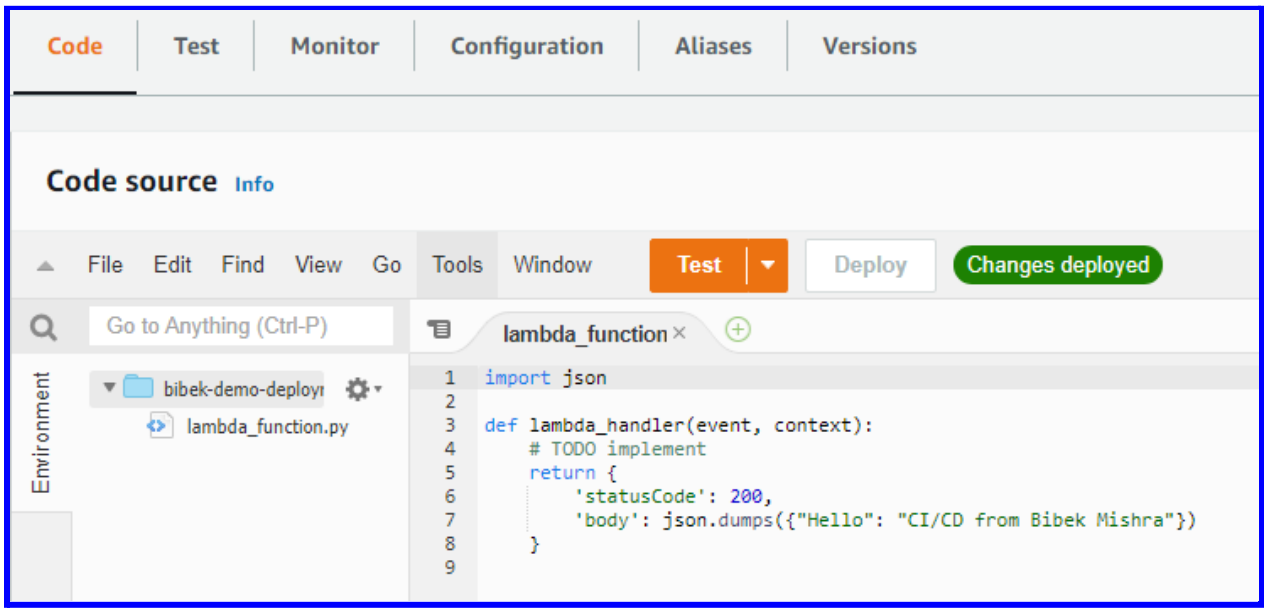
Workflow triggered itself



Update function completed



If we see the code of lambda function it has been changed successfully



In this way we can build a pipeline through Github Actions which can update the lambda function whenever any changes are pushed to the main branch