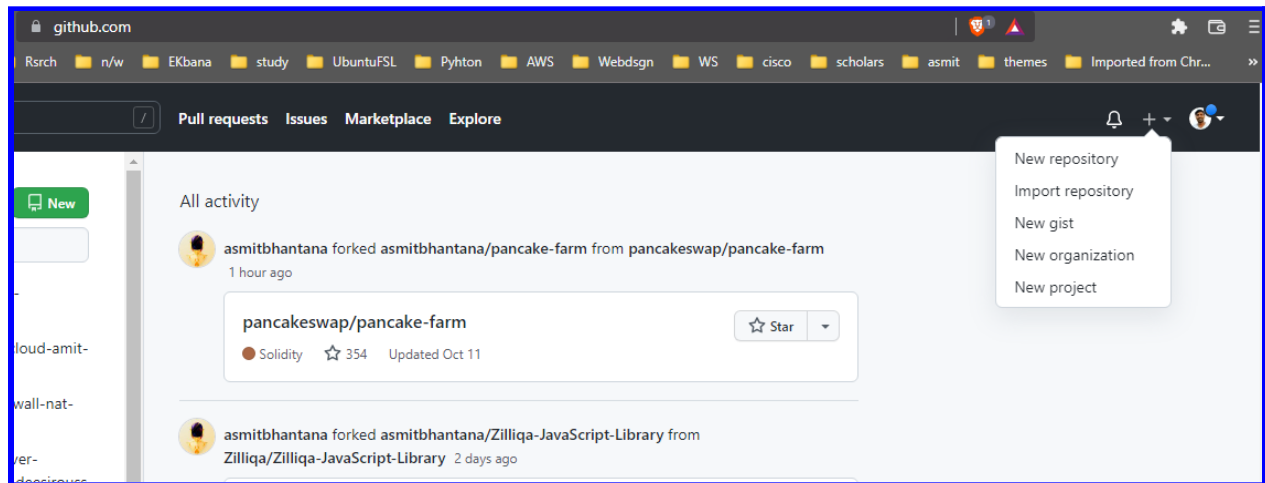


Integrate both these scripts with one of Jenkins, Github Actions, CircleCI or TravisCI


I have used **GitHub Actions**

Creating new repository for Lambda CI/CD



Name - aws-cicd, Private Repo

Owner *

 deesirouss ▾


/

Repository name *


aws-cicd ✓

Great repository names are short and memorable. Need inspiration? How about [musical-octo-memory](#)?

Description (optional)

☐  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Creating Secret key for AWS Access key and Secret key

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

Options

Manage access

Security & analysis

Webhooks

Notifications

Integrations

Deploy keys

Actions

Secrets

Actions

Dependabot

Actions secrets

New repository secret

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

There are no secrets for this repository.

Encrypted secrets allow you to store sensitive information, such as access tokens, in your repository.

Actions secrets / New secret

Name

AWS_ACCESS_KEY_ID

Value

Add secret

Actions secrets / New secret

Name

AWS_SECRET_ACCESS_KEY

Value

[Redacted value]

Add secret

We have two Secrets

Actions secrets

New repository secret

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).



AWS_ACCESS_KEY_ID

Updated 1 minute ago

Update

Remove



AWS_SECRET_ACCESS_KEY

Updated in 15 seconds

Update

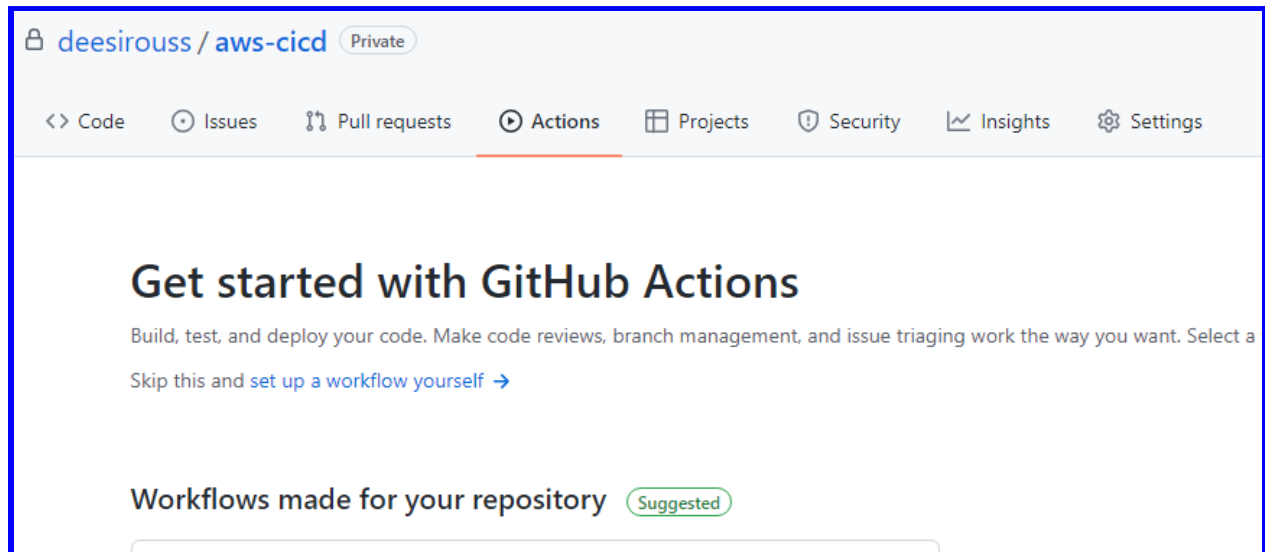
Remove

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

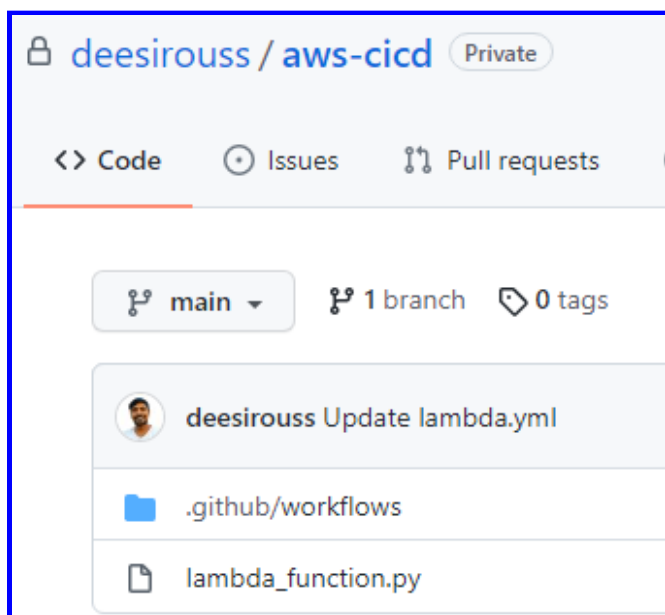
    #AWS credentials
    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
        aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
        # TODO Change your AWS region here!
        aws-region: us-east-2

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

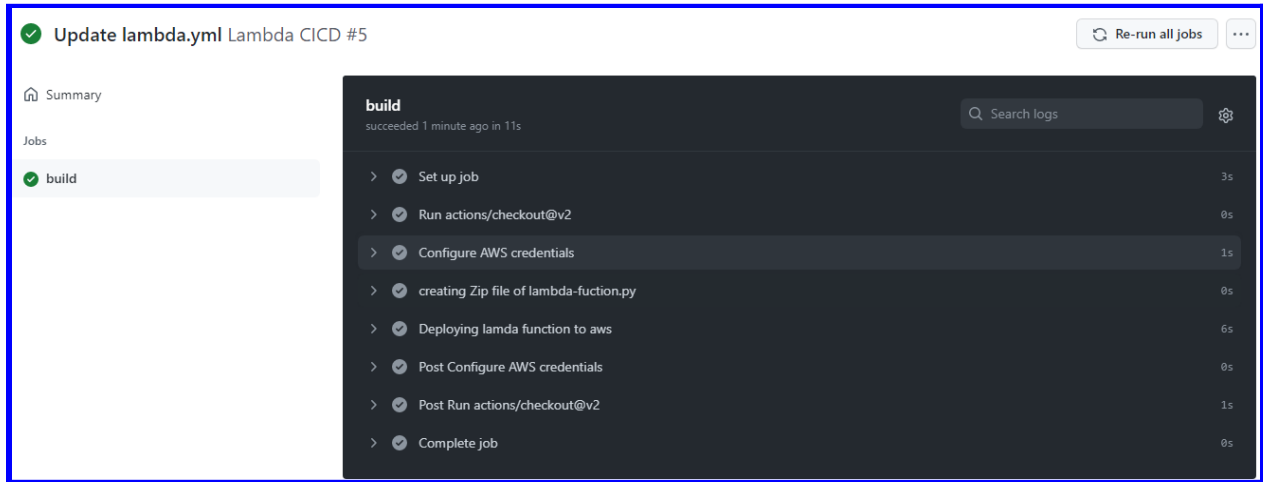
    - name: Deploying lambda function to aws
      run: |
        aws lambda create-function --function-name bibek-demo-deployment \
        --runtime python3.9 --zip-file fileb://lambda_function.zip \
        --role arn:aws:iam::949263681218:role/service-role/bibek-api-gw-default-role-mof54wmn \
      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

```

We have two files in our repository



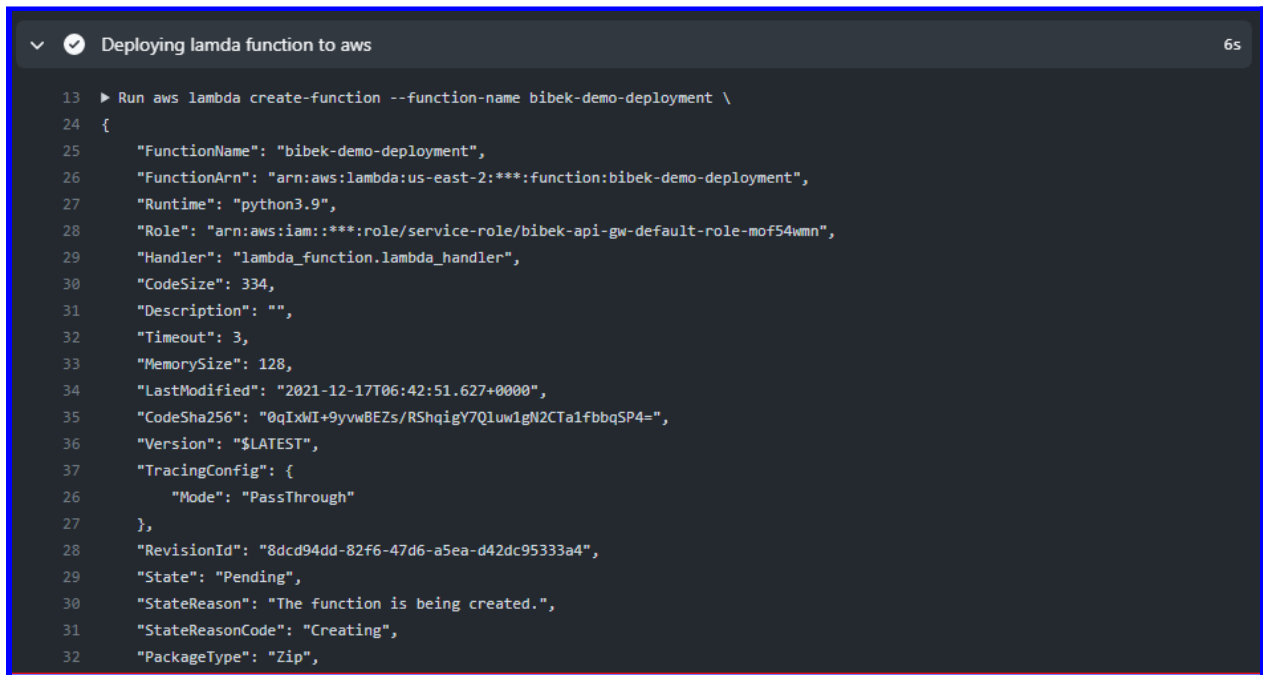
If we see the build result, all jobs have been executed successfully



The screenshot shows a GitHub Actions workflow named "Update lambda.yml" with ID "Lambda CICD #5". The workflow is in a "Completed" state, indicated by a green checkmark. The "Summary" tab is selected, showing a list of jobs. The "build" job is highlighted, showing a green checkmark and a duration of "succeeded 1 minute ago in 11s". The "build" job details are expanded, showing a list of steps:

- Set up job (3s)
- Run actions/checkout@v2 (0s)
- Configure AWS credentials (1s)
- creating Zip file of lambda-fuction.py (0s)
- Deploying lamda function to aws (6s)
- Post Configure AWS credentials (0s)
- Post Run actions/checkout@v2 (1s)
- Complete job (0s)

Deploying Lambda function to aws Job details



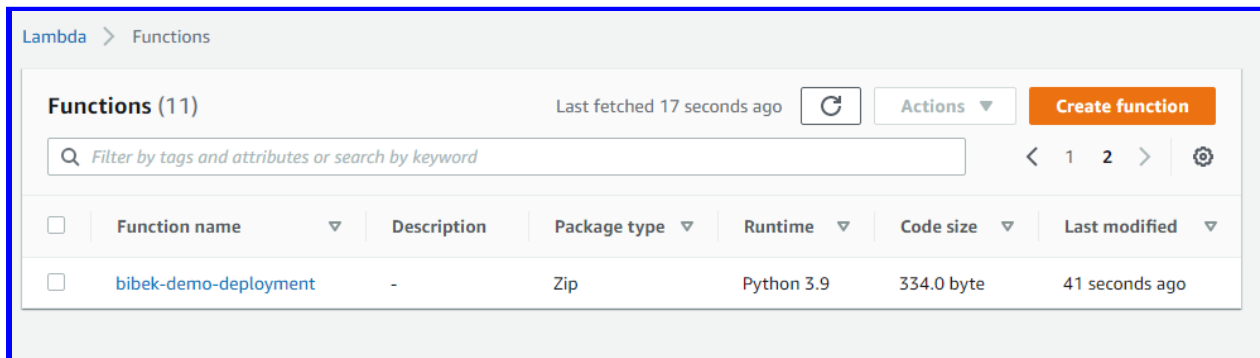
The screenshot shows the details of the "Deploying lamda function to aws" job, which took 6s to complete. The job is in a "Completed" state, indicated by a green checkmark. The command being executed is:

```
Run aws lambda create-function --function-name bibek-demo-deployment \
```

The output shows the details of the function being created:

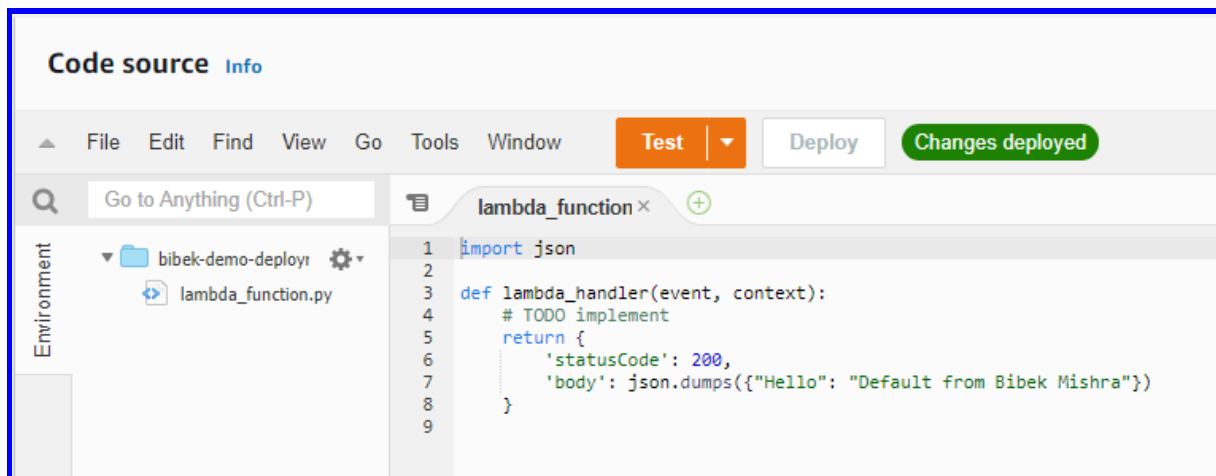
```
{
  "FunctionName": "bibek-demo-deployment",
  "FunctionArn": "arn:aws:lambda:us-east-2:::function:bibek-demo-deployment",
  "Runtime": "python3.9",
  "Role": "arn:aws:iam:::role/service-role/bibek-api-gw-default-role-mof54wmn",
  "Handler": "lambda_function.lambda_handler",
  "CodeSize": 334,
  "Description": "",
  "Timeout": 3,
  "MemorySize": 128,
  "LastModified": "2021-12-17T06:42:51.627+0000",
  "CodeSha256": "0qIxnI+9yvwBEZs/RShqigV7Q1uw1gN2CTa1fbbqSP4=",
  "Version": "$LATEST",
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "RevisionId": "8dcd94dd-82f6-47d6-a5ea-d42dc95333a4",
  "State": "Pending",
  "StateReason": "The function is being created.",
  "StateReasonCode": "Creating",
  "PackageType": "Zip",
```

We can see the new lambda function has been deployed recently



The screenshot shows the AWS Lambda console 'Functions' page. At the top, it says 'Functions (11)' and 'Last fetched 17 seconds ago'. There is a search bar and a 'Create function' button. Below is a table with columns: Function name, Description, Package type, Runtime, Code size, and Last modified. One function is listed: 'bibek-demo-deployment' with a description of '-', package type 'Zip', runtime 'Python 3.9', code size '334.0 byte', and last modified '41 seconds ago'.

| Function name | Description | Package type | Runtime | Code size | Last modified |
|-----------------------|-------------|--------------|------------|------------|----------------|
| bibek-demo-deployment | - | Zip | Python 3.9 | 334.0 byte | 41 seconds ago |



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


The screenshot displays the GitHub Actions interface for a workflow named 'Update lambda.yml' (run ID: Lambda CICD #7). The workflow status is 'Completed' with a green checkmark. The 'build' job is highlighted in the 'Jobs' list on the left. The main panel shows the 'build' job details, indicating it 'succeeded 26 seconds ago in 10s'. A list of steps is shown, all of which completed successfully:


| Step | Duration |
|--|----------|
| Set up job | 3s |
| Run actions/checkout@v2 | 1s |
| Configure AWS credentials | 0s |
| creating Zip file of lambda-fuction.py | 0s |
| Deploying lamda function to aws | 6s |
| Post Configure AWS credentials | 0s |
| Post Run actions/checkout@v2 | 0s |
| Complete job | 0s |

Update function completed

build

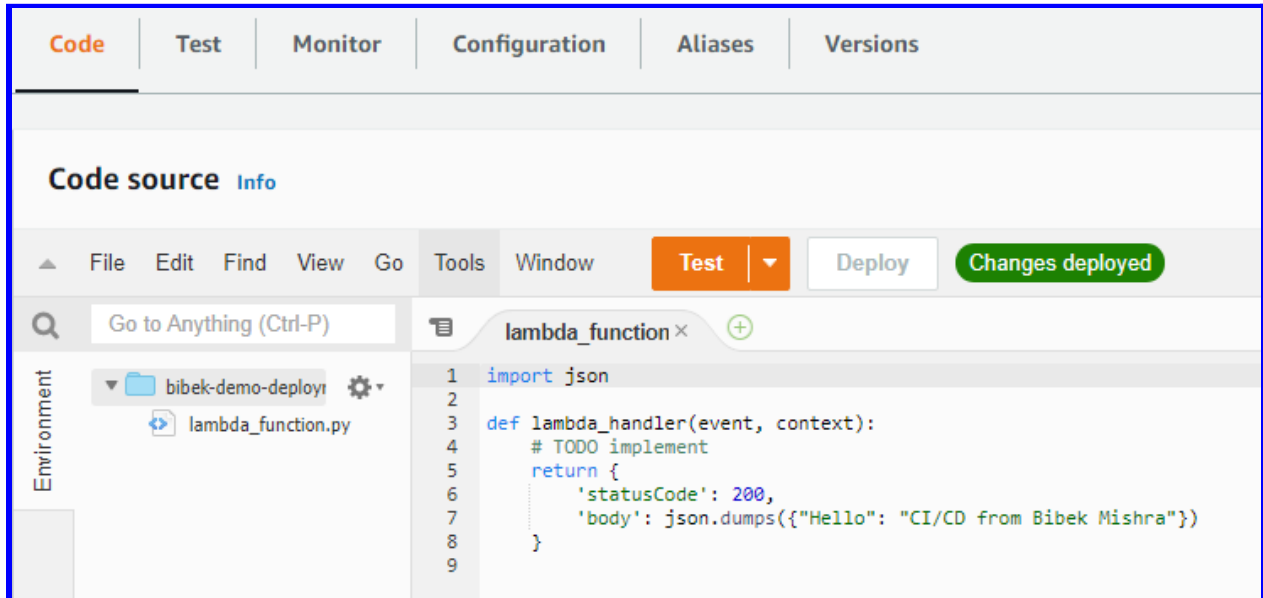
succeeded 26 seconds ago in 10s

>  creating Zip file of lambda-fuction.py

▼  Deploying lamda function to aws

```
1 ▶ Run aws lambda update-function-code --function-name bibek-demo-deployment \  
10 {  
11     "FunctionName": "bibek-demo-deployment",  
12     "FunctionArn": "arn:aws:lambda:us-east-2::*:function:bibek-demo-deployment",  
13     "Runtime": "python3.9",  
14     "Role": "arn:aws:iam::*:role/service-role/bibek-api-gw-default-role-mof54wmn",  
15     "Handler": "lambda_function.lambda_handler",  
16     "CodeSize": 334,  
17     "Description": "",  
18     "Timeout": 3,  
19     "MemorySize": 128,  
20     "LastModified": "2021-12-17T06:51:47.000+0000",  
21     "CodeSha256": "Oy72nJGrU91D2Ve56ATV10eveP6pqeM7ZA6rEJ4D6Z8=",  
22     "Version": "$LATEST",  
23     "TracingConfig": {  
24         "Mode": "PassThrough"  
25     },  
26     "RevisionId": "ab0d8b78-1afd-47ab-98fd-f8b72fc244fa",  
27     "State": "Active",  
28     "LastUpdateStatus": "InProgress",  
29     "LastUpdateStatusReason": "The function is being created.",  
30     "LastUpdateStatusReasonCode": "Creating",
```

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