Lambda CICD User Guide

Github Actions is a Continuous Integration and Continuous Deployment (CI/CD) tool that provides a powerful and flexible platform for automating software development workflows.

AWS SAM makes it easier to create continuous integration and deployment (CI/CD) pipelines with Github Actions. SAM is an open-source framework developed by AWS that simplifies the deployment and management of serverless applications. SAM extends AWS CloudFormation to provide a simplified syntax specifically designed for serverless resources. GitHub Actions is a CI/CD platform provided by GitHub that allows you to automate various workflows, tasks, and processes directly within your GitHub repository.

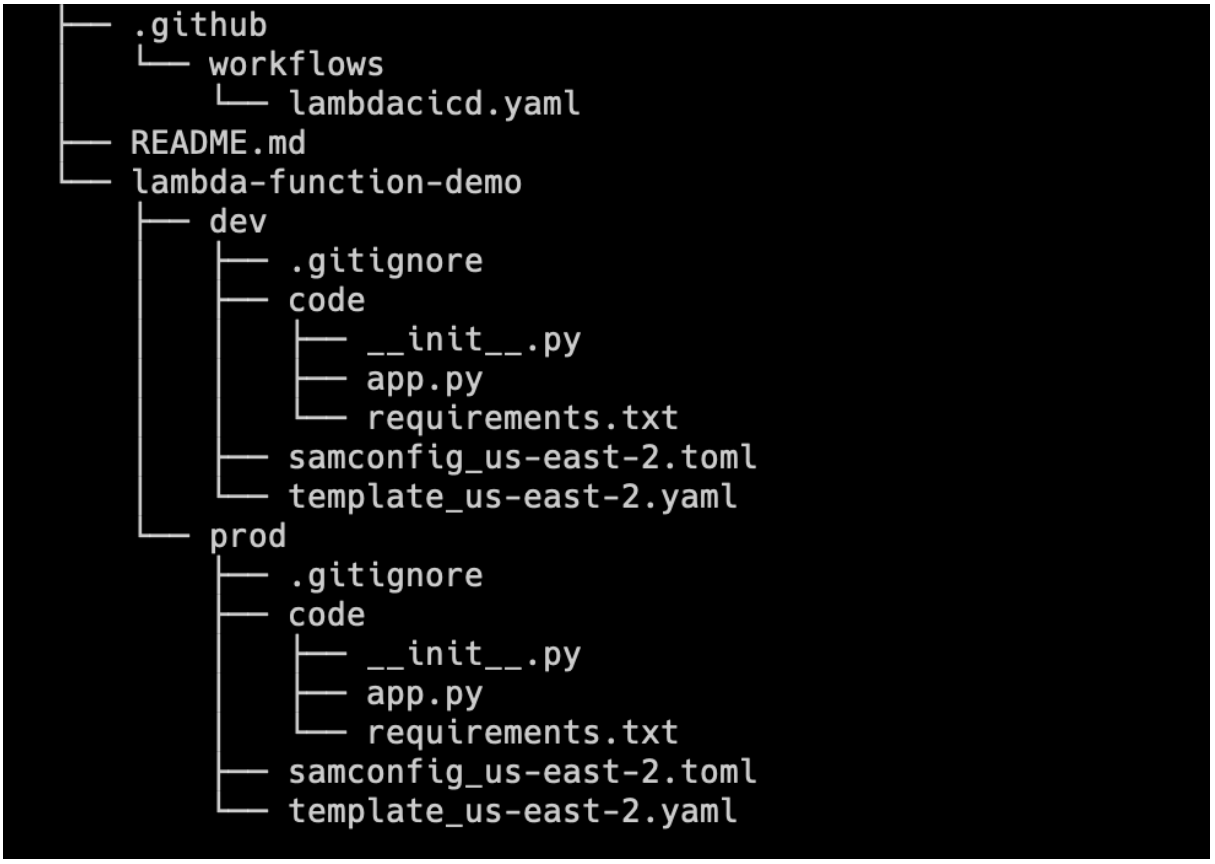
To onboard a new Lambda function in Github Actions CI/CD, follow these steps:

Prerequisites:

- The IAM Role for Lambda should be created with all required permissions by the DevOps team, and then they will provide the ARN to use in the SAM template.
- If you are planning to use a Layer, it should be created beforehand for the new Lambda.

Steps:

- Refer Github repository https://github.com/prodigal-tech/lambda-cicd-template which contains templates of Github Actions Workflow file, SAM template file, SAM config file and folder name "**code**" where code need to be placed. Take reference from these files and add them in your respective github repos with same hierarchy.



- Use below `lambdacicd.yaml` cicd file to create GitHub Action workflow. Approvars need to be added in this file at approval step. We can avoid approval step if require.

▼ **lambdacicd.yaml**

```yaml
name: Deploy Lambda
on:
  workflow_dispatch:
    inputs:
      region:
        description: 'Region'
        type: choice
        required: true
        default: 'us-east-2'
        options:
        - us-east-2
        - ca-central-1
        - eu-west-2
      deploy_path:
```

```
        description: 'Path of lambda template'
        type: string
        required: true
        default: 'lambda-cicd-prod'
jobs:
  build-deploy:
    runs-on: ubuntu-20.04
    permissions:
        issues: write
        id-token: write
        contents: read
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v4
        with:
          python-version: '3.9'
      - uses: actions/setup-go@v5
        with:
          go-version: '1.18'
      - name: Configure private token
        env:
          GH_ACCESS_TOKEN: ${{ secrets.GLOBAL_GITHUB_HTTPS
        run: |
          git config --global url."https://${GH_ACCESS_TOK
      - uses: actions/setup-node@v3
        with:
          node-version: 18
      - uses: aws-actions/setup-sam@v2
      - uses: aws-actions/configure-aws-credentials@v2
        with:
          role-to-assume: arn:aws:iam::532968567499:role/O
          aws-region: ${{ inputs.region }}
      # sam build
      - name: SAM Build
        run: |
          cd ${{ github.workspace }}/${{ inputs.deploy_pat
          aws codeartifact login --tool pip --domain prodi
          sam build -t template_${{ inputs.region }}.yaml
```

```
      - name: Approval
        uses: trstringer/manual-approval@v1
        with:
          secret: ${{ github.TOKEN }}
          approvers: RahulGP14
          timeout-minutes: 10
          minimum-approvals: 1
      # sam deploy
      - name: SAM Deploy
        run: |
          cd ${{ github.workspace }}/${{ inputs.deploy_pat
          sam deploy --config-file samconfig_${{ inputs.re
```

- Update the SAM config file `samconfig_us-east-2.toml` with the `stack_name` parameter in the format “{FunctionName}-stack”

▼ **samconfig_us-east-2.toml**

```
version = 0.1
[default]
[default.deploy]
[default.deploy.parameters]
stack_name = "Ohlambdademodev-stack" #Update Cloudformatio
s3_bucket = "sam-lambda-cicd-bucket"
region = "us-east-2" # Change region as per requirement
confirm_changeset = true
capabilities = "CAPABILITY_IAM"
disable_rollback = true
image_repositories = []
```

- All Lambda function configurations are present in `template_us-east-2.yaml` Modify this file as per your requirements. Remove or comment sections in the file if not required (e.g., Remove `VpcConfig` section if access to DB or ElasticCache is not required, Remove Events section if a trigger is not required).

▼ **template_us-east-2.yaml**

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  lambda-cicd

  Sample SAM Template for lambda-cicd

Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: OhLambdaCICDFunctionDemoDevSG # Security
      GroupDescription: Allow HTTP traffic to the host
      VpcId: vpc-4645182e
      SecurityGroupEgress:
      - IpProtocol: "-1"
        CidrIp: 0.0.0.0/0
  LambdaCICDFunctionDev:
    Type: AWS::Serverless::Function
    Properties:
      FunctionName: OhLambdaCICDFunctionDemoDev # Function
      CodeUri: code/
      Handler: lambda_function.lambda_handler # Lambda Han
      Runtime: python3.9 # Runtime Programming Language
      Tracing: Active
      Timeout: 20  # Lambda Timeout in seconds
      MemorySize: 256  # Memory size
      Role: arn:aws:iam::532968567499:role/< Lambda Role >
      Architectures:
        - x86_64
      Environment:
        Variables:
          ENVIRONMENT_VERIABLE: Environment-value # Add en
      VpcConfig:
        SecurityGroupIds:
          - Ref: SecurityGroup
        SubnetIds:
          - subnet-01f783605e41b71ba
```

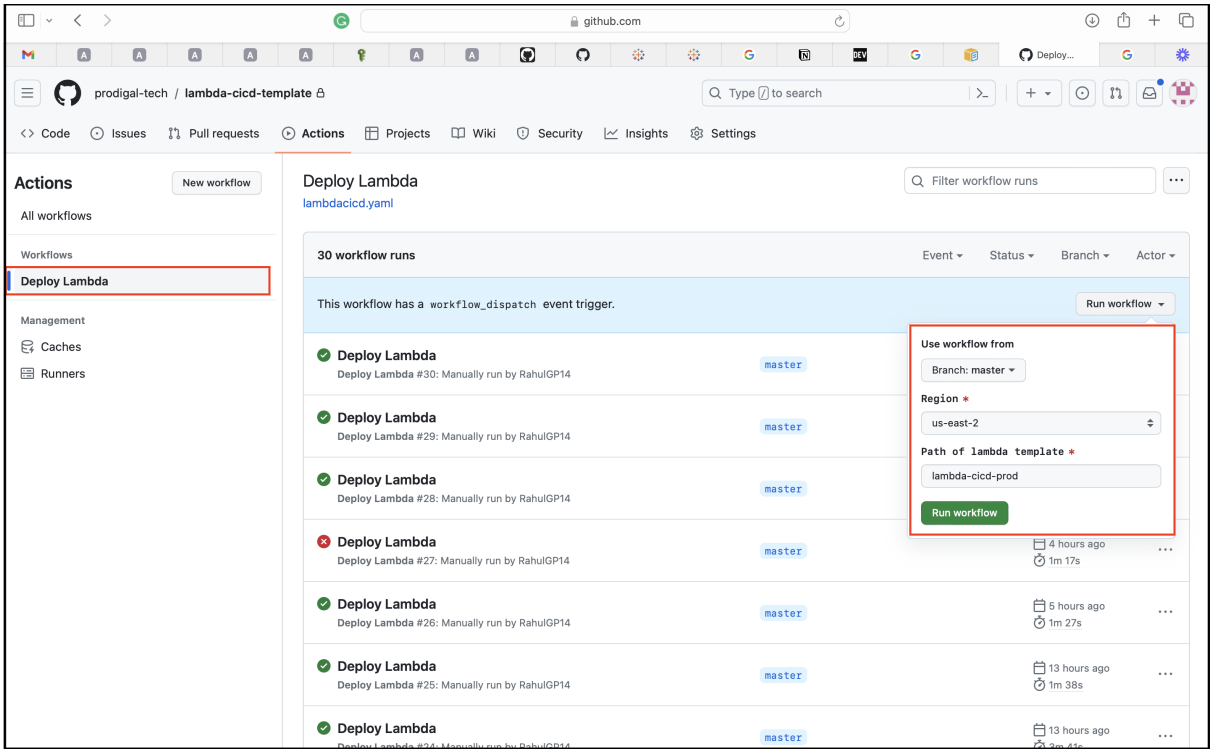
```
      Events:
          EventBridgeCron:
            Type: Schedule
            Properties:
              Schedule: cron(0 14 * * ? *) # Eventbridge f
              Name: LambdaCICDFunctionDev-Schedule # Event
              Description: Example schedule
              Enabled: true
          Sqstrigger:
            Type: SQS
            Properties:
              Queue: arn:aws:sqs:us-east-2:532968567499:la
              BatchSize: 10
              Enabled: true
          Snstrigger:
            Type: SNS
            Properties:
              Topic: arn:aws:sns:us-east-2:532968567499:Pr
      FileSystemConfigs:
        - Arn: >-
            arn:aws:elasticfilesystem:us-east-2:5329685674
          LocalMountPath: /mnt/efs # Mount path
      Layers:
        - arn:aws:lambda:us-east-2:532968567499:layer:libs

Outputs:
  LambdaCICDFunctionDev:
    Description: "us-east-2 Cupid Cleanup Lambda Function A
    Value: !GetAtt LambdaCICDFunctionDev.Arn
```

- Update the following parameters with your desired configurations in `samconfig_us-east-2.toml` and `template_us-east-2.yaml`, or else deployment will fail.

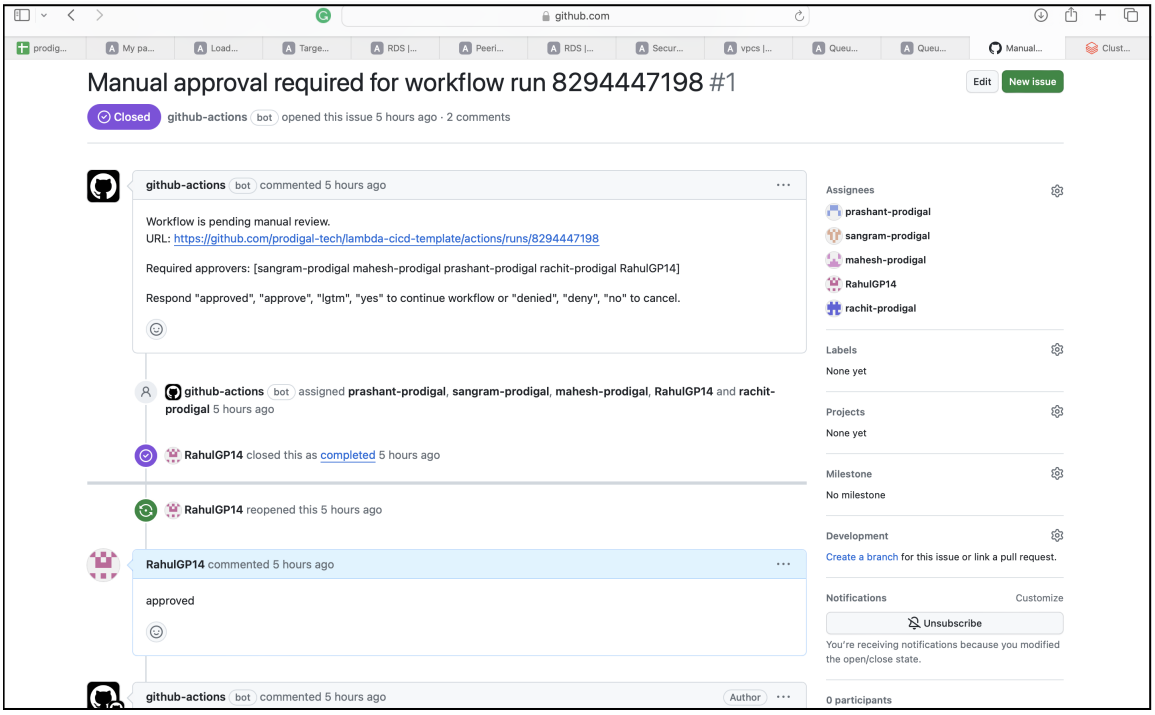
```
samconfig:
- stack_name

template:
```

```
- SecurityGroup.GroupName
- Role
- EventBridgeCron.Properties.Name
- Sqstrigger.Properties.Queue
- Snstrigger.Properties.Topic
```

- This CI/CD pipeline is created for the `us-east-2` region. To deploy it to another region, create a new `samconfig_<Region Name>.toml` and `template_<Region Name>.yaml` file and update the region in the `samconfig` file in same directory.
- Now, when you want to deploy the Lambda function, go to the Actions Tab in the repo and click on `Deploy Lambda` in the left section. Check on `Run workflow`, change the fields, and click the `Run Workflow` button at the bottom.

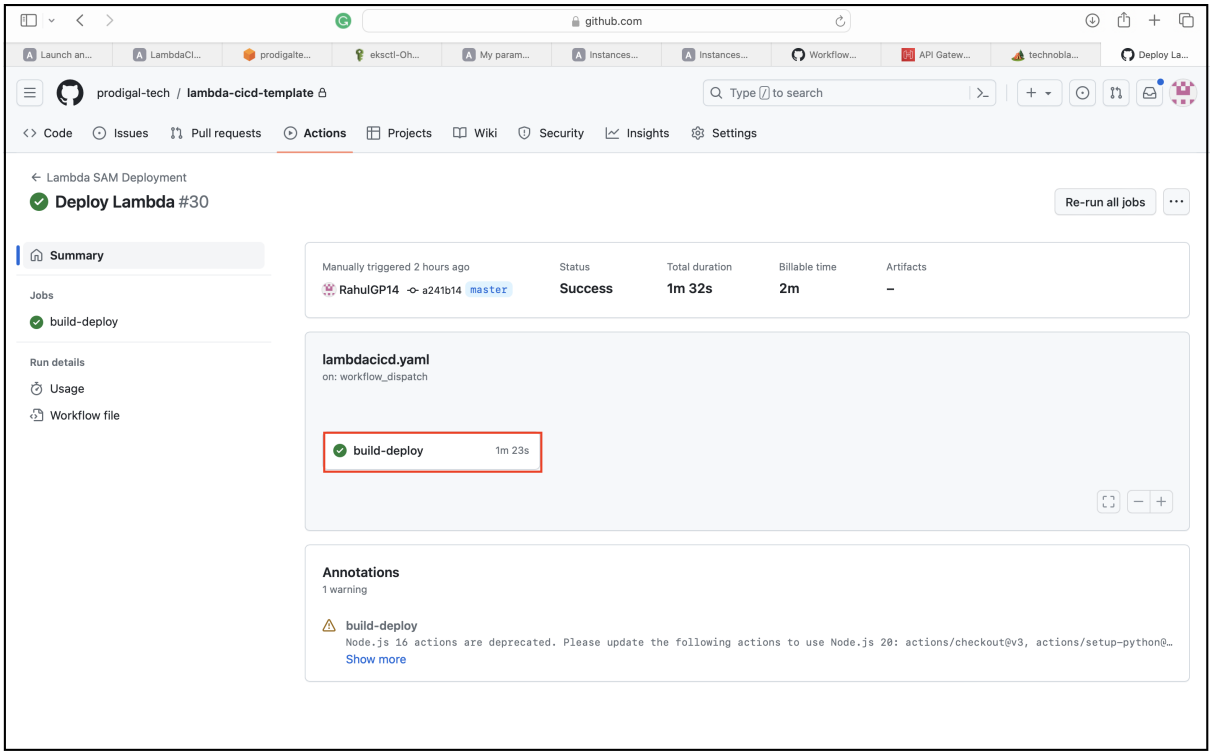


- This will trigger a Github action run and will have three steps:
  1. SAM build: This step will build the SAM template
  2. Approval: Create an issue in repo and assign it to respective leads to approve the build.



1. Once any one lead comments "approved"/ "approve"/ "lgtm"/ "yes" on the issue, the workflow will continue to SAM deploy step and will deploy the lambda to AWS.

There is a timeout of 10 mins on approvals, if the lead does not approves within this duration, the action will timeout.



- If required, we can also add multiple approvers. So, if any one of the approvers approves within 10 minutes, then only the lambda gets deployed or updated.

If there is no to update to code or lambda config, then deployment will fail.