Create a bash script to deploy your lambda functions
Create a bash script to deploy your react app to s3
Integrate both these scripts with one of jenkins,Github actions,CircleCI or TravisCI.

We created the following files for our lambda function

```
samana@samana:~/lambda_functions$ cat hello.py
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps({"Hello":"Default"})
    }
samana@samana:~/lambda_functions$ cat hello_dynamic.py
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps({"Hello": event["rawPath"].strip('/')})
    }
samana@samana:~/lambda_functions$
```

We create a script to update our lambda functions as follows:

Open *lambda.sh

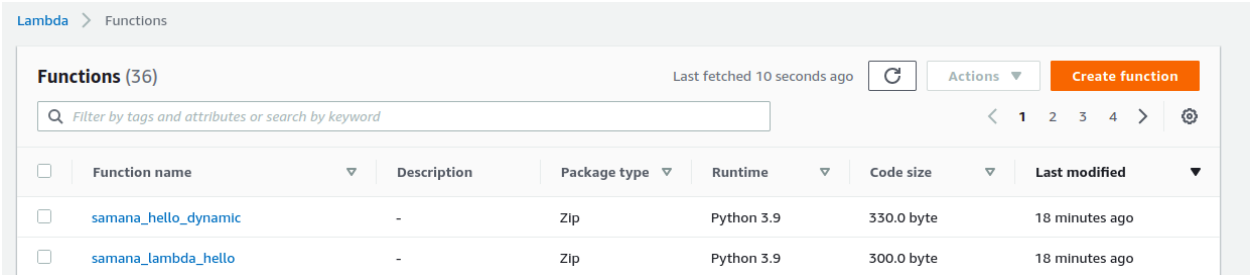
```
#!/bin/bash

cd /home/samana/lambda_functions

# Create zip packages for the deployment
zip package.zip hello.py
zip package_dynamic.zip hello_dynamic.py

# For first function redeploy the zip package
aws lambda update-function-code  --function-name samana_lambda_hello \
--zip-file fileb://package.zip


# For second function
aws lambda update-function-code  --function-name samana_hello_dynamic \
--zip-file fileb://package_dynamic.zip

```

Running the script:

```
samana@samana:~$ ./lambda.sh
updating: hello.py (deflated 18%)
updating: hello_dynamic.py (deflated 19%)
{
    "FunctionName": "samana_lambda_hello",
    "FunctionArn": "arn:aws:lambda:us-east-1:949263681218:function:samana_lambda_hello",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::949263681218:role/service-role/samana_role_hello",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 300,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2021-12-17T10:38:37.000+0000",
    "CodeSha256": "Ndxc2A5z8XyLk9Fu359WeMXJ4o4Hplq5Z9PqQMz0n1U=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "d5526cee-a87d-417c-b97a-d88e7db6bef4",
    "State": "Active",
    "LastUpdateStatus": "InProgress",
    "LastUpdateStatusReason": "The function is being created.",
    "LastUpdateStatusReasonCode": "Creating",
    "Architectures": [
:...skipping...
{
    "FunctionName": "samana_lambda_hello",
    "FunctionArn": "arn:aws:lambda:us-east-1:949263681218:function:samana_lambda_hello",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::949263681218:role/service-role/samana_role_hello",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 300,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2021-12-17T10:38:37.000+0000",
    "CodeSha256": "Ndxc2A5z8XyLk9Fu359WeMXJ4o4Hplq5Z9PqQMz0n1U="
```

```
{
    "FunctionName": "samana_hello_dynamic",
    "FunctionArn": "arn:aws:lambda:us-east-1:949263681218:function:samana_hello_dynamic",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::949263681218:role/service-role/samana_role_hello",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 330,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2021-12-17T10:38:45.000+0000",
    "CodeSha256": "/SooYvT/nSoeqS9KZPTnRF/3oOoYYLLDyxvdR6/LVx8=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "70bb923c-f15e-4477-ab58-ba6302de25b2",
    "State": "Active",
    "LastUpdateStatus": "InProgress",
    "LastUpdateStatusReason": "The function is being created.",
    "LastUpdateStatusReasonCode": "Creating",
    "Architectures": [
        "x86_64"
    ]
}
samana@samana:~$
```


Result: created lambda functions.



Now,we have our directory for react application with the following contents

```
samana@samana:~$ ls demoreact/
node_modules  package.json  package-lock.json  public  README.md  src
samana@samana:~$
```

We build the react application and upload the build artifacts to our bucket using the following script.

Open

reactapp_s3.sh

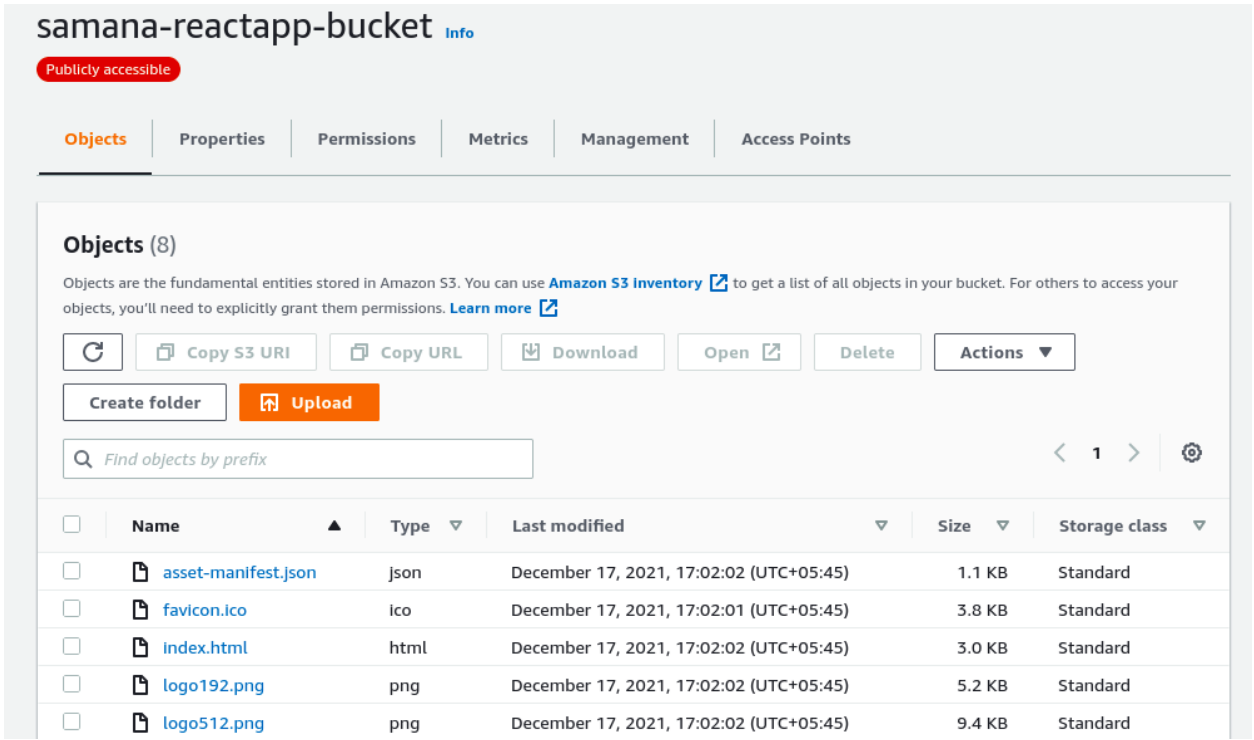
```
#!/bin/bash

cd ~/demoreact

# build the static react application
npm run build

cd ~/demoreact/build

# Update the bucket with new build artifacts
aws s3 sync ./ s3://samana-reactapp-bucket

```

Running the script:

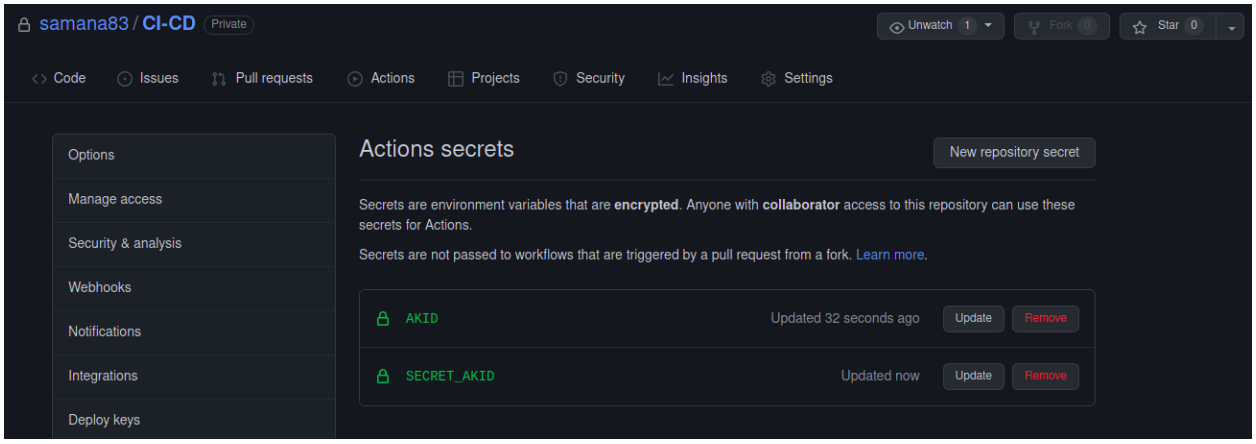
```
samana@samana:~$ chmod +x reactapp_s3.sh
samana@samana:~$ ./reactapp_s3.sh

> demoreact@0.1.0 build /home/samana/demoreact
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  41.34 KB  build/static/js/2.972298a5.chunk.js
  1.62 KB   build/static/js/3.c1c180f4.chunk.js
  1.16 KB   build/static/js/runtime-main.4c78a30b.js
  574 B     build/static/js/main.caa169e1.chunk.js
  548 B     build/static/css/main.b99c2553.chunk.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build
```

```
The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

upload: ./favicon.ico to s3://samana-reactapp-bucket/favicon.ico
upload: static/js/2.972298a5.chunk.js.LICENSE.txt to s3://samana-reactapp-bucket/static/js/2.972298a5.chunk.js.LICENSE.txt
upload: ./robots.txt to s3://samana-reactapp-bucket/robots.txt
upload: ./asset-manifest.json to s3://samana-reactapp-bucket/asset-manifest.json
upload: static/css/main.b99c2553.chunk.css.map to s3://samana-reactapp-bucket/static/css/main.b99c2553.chunk.css.map
upload: ./logo512.png to s3://samana-reactapp-bucket/logo512.png
upload: static/css/main.b99c2553.chunk.css to s3://samana-reactapp-bucket/static/css/main.b99c2553.chunk.css
upload: ./manifest.json to s3://samana-reactapp-bucket/manifest.json
upload: ./index.html to s3://samana-reactapp-bucket/index.html
upload: ./logo192.png to s3://samana-reactapp-bucket/logo192.png
upload: static/js/3.c1c180f4.chunk.js to s3://samana-reactapp-bucket/static/js/3.c1c180f4.chunk.js
upload: static/js/main.caa169e1.chunk.js to s3://samana-reactapp-bucket/static/js/main.caa169e1.chunk.js
upload: static/js/main.caa169e1.chunk.js.map to s3://samana-reactapp-bucket/static/js/main.caa169e1.chunk.js.map
upload: static/js/runtime-main.4c78a30b.js to s3://samana-reactapp-bucket/static/js/runtime-main.4c78a30b.js
upload: static/js/runtime-main.4c78a30b.js.map to s3://samana-reactapp-bucket/static/js/runtime-main.4c78a30b.js.map
upload: static/js/3.c1c180f4.chunk.js.map to s3://samana-reactapp-bucket/static/js/3.c1c180f4.chunk.js.map
upload: static/js/2.972298a5.chunk.js to s3://samana-reactapp-bucket/static/js/2.972298a5.chunk.js
upload: static/media/logo.bbe07cfb.png to s3://samana-reactapp-bucket/static/media/logo.bbe07cfb.png
upload: static/js/2.972298a5.chunk.js.map to s3://samana-reactapp-bucket/static/js/2.972298a5.chunk.js.map
samana@samana:~$
```

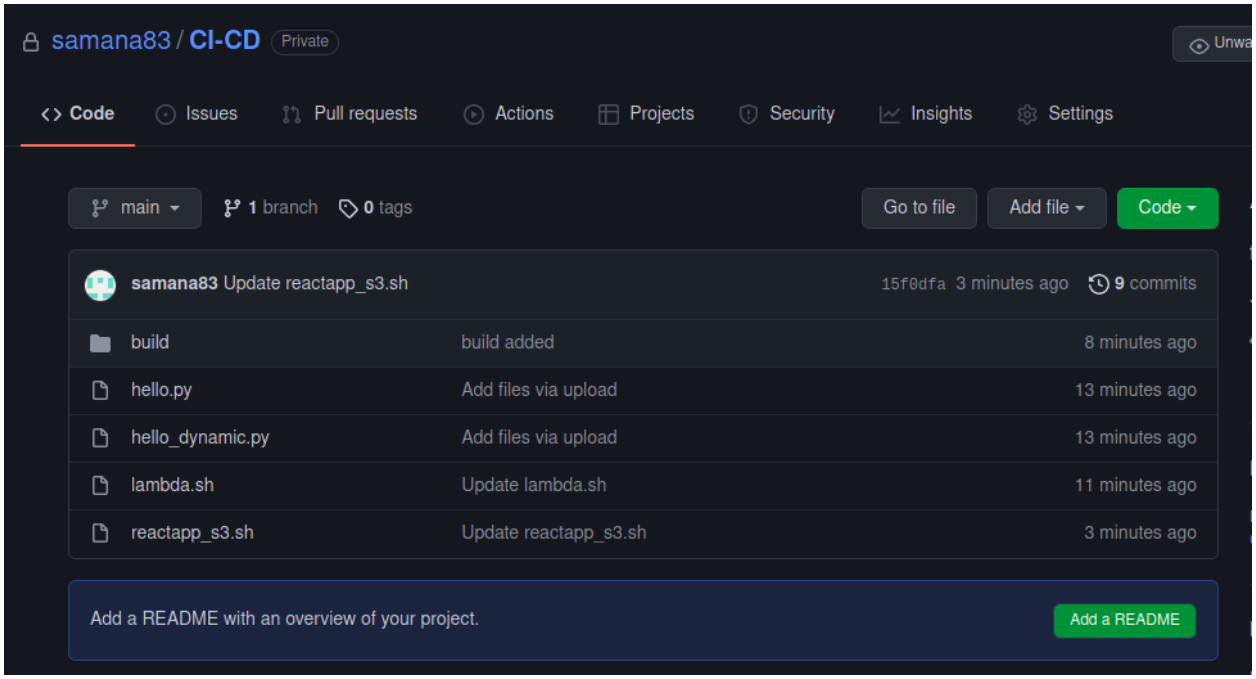
Bucket was updated as follows:



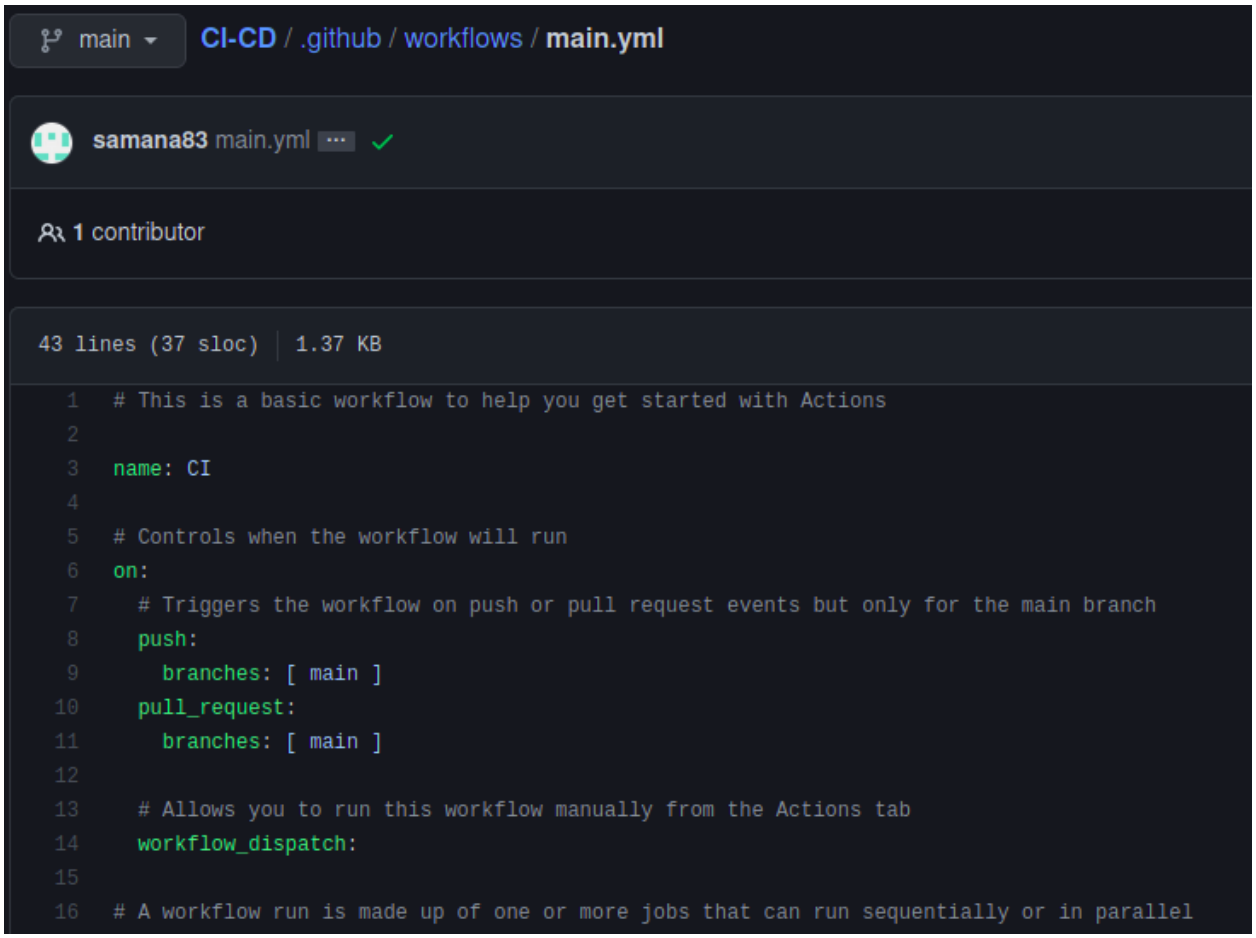
Now for integrating both of the above scripts we use Github actions
First we set the access key id and secret access key for our aws account as actions secret



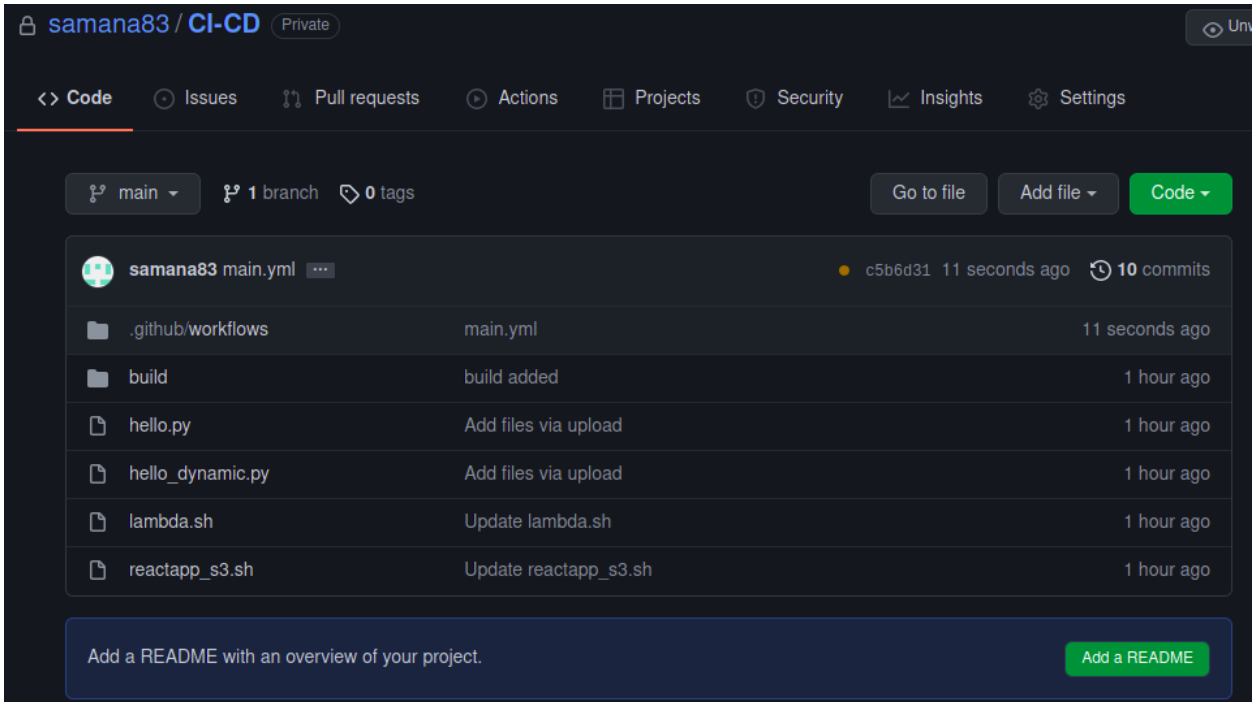
Then we upload the following files (our scripts, our python files for lambda function and the build folder for our react application)



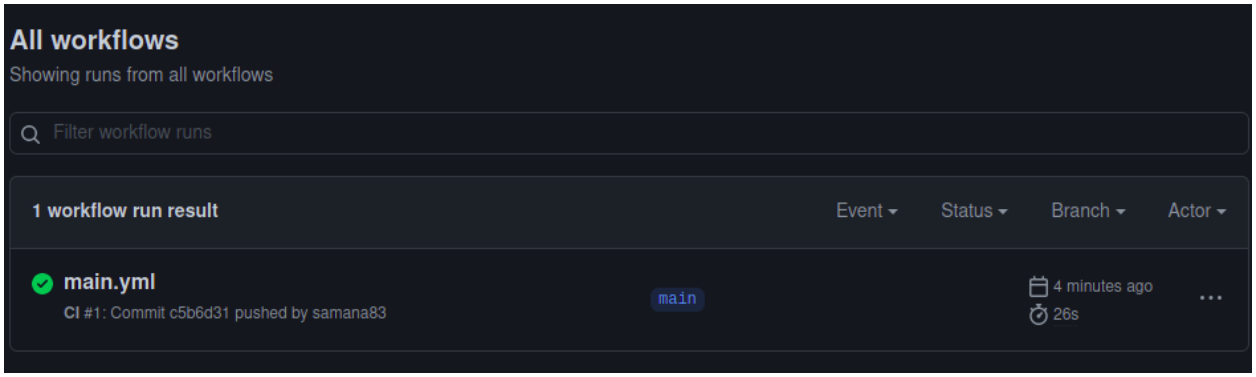
Now we create a main.yml workflow file as follows:



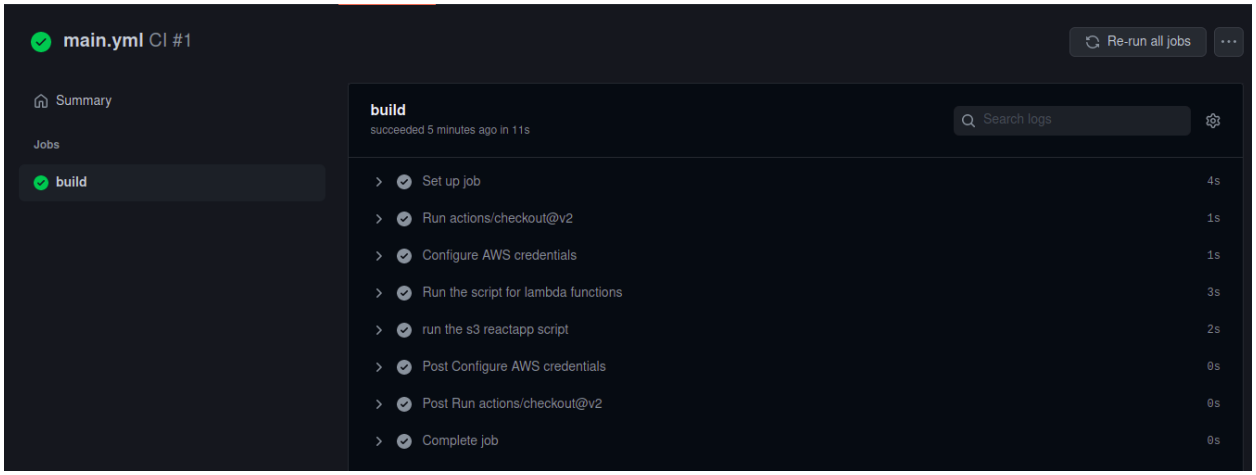
```yaml
  workflow_dispatch:

# 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

      # Runs a single command using the runners shell
      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AKID }}
          aws-secret-access-key: ${{ secrets.secret_AKID}}
          aws-region: us-east-1
      # Runs a set of commands using the runners shell
      - name: Run the script for lambda functions
        run: |
          chmod +x lambda.sh
          ./lambda.sh
      - name: run the s3 reactapp script
        run: |
          chmod +x reactapp_s3.sh
          ./reactapp_s3.sh
```



The workflow file was run:

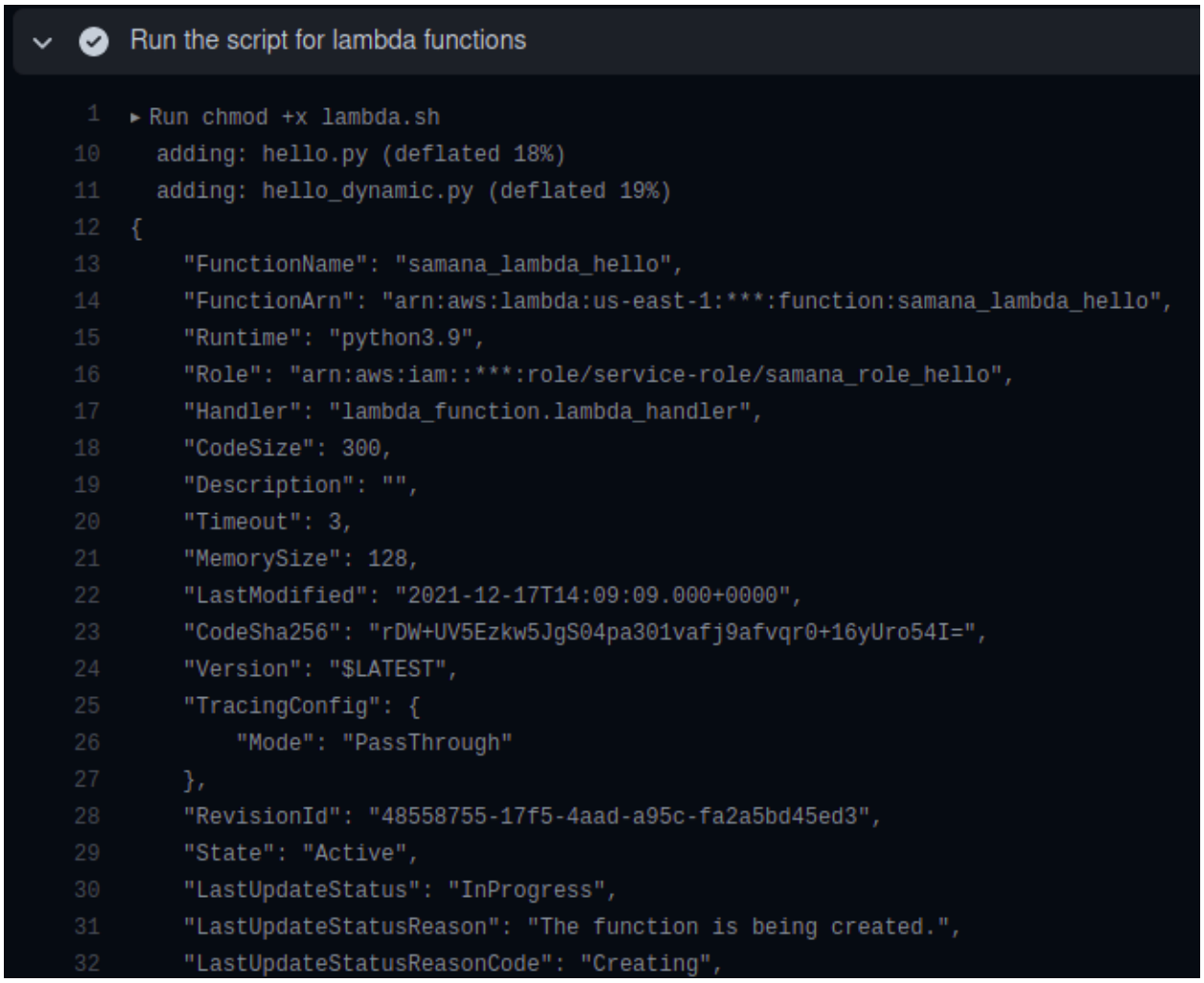


The following build steps were successful



Configuring aws credentials:

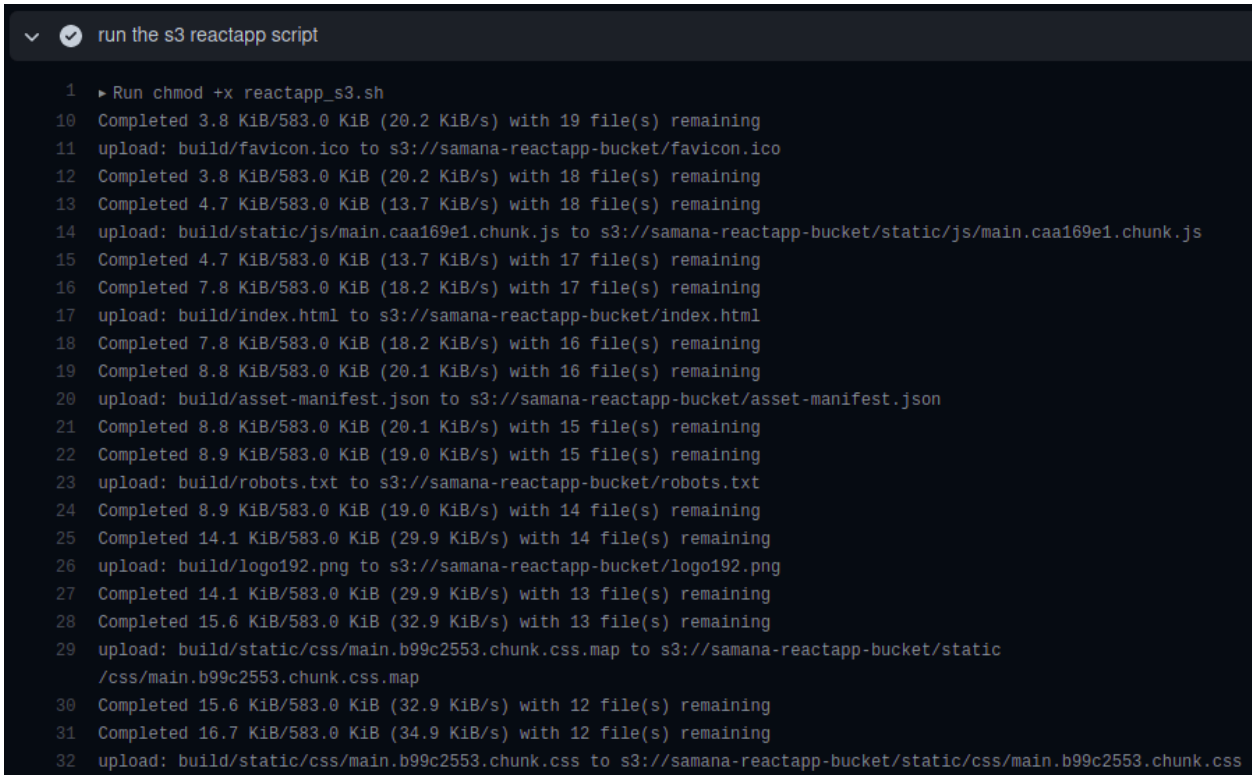
The script for lambda function was also run successfully.



Run the script for lambda functions

```
    ]
}
{
    "FunctionName": "samana_hello_dynamic",
    "FunctionArn": "arn:aws:lambda:us-east-1:***:function:samana_hello_dynamic",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::***:role/service-role/samana_role_hello",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 330,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2021-12-17T14:09:10.000+0000",
    "CodeSha256": "k+NE1s5N9/j5JylDYiVNd9ziSn208gUDFfAQ9cf2IGk=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "d7ef6b5a-cb54-43d5-ae67-29089265e76f",
    "State": "Active",
    "LastUpdateStatus": "InProgress",
    "LastUpdateStatusReason": "The function is being created.",
    "LastUpdateStatusReasonCode": "Creating",
    "Architectures": [
        "x86_64"
    ]
```

The script for our react application was also run successfully as follows:



Result: The API gateway's invoke URL showed our desired resonse

Result: Our react s3 application: