

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  [icon]  *lambda.sh

1 #!/bin/bash
2
3 cd /home/samana/lambda_functions
4
5 # Create zip packages for the deployment
6 zip package.zip hello.py
7 zip package_dynamic.zip hello_dynamic.py
8
9 # For first function redeploy the zip package
10 aws lambda update-function-code --function-name samana_lambda_hello \
11 --zip-file fileb://package.zip
12
13
14 # For second function
15 aws lambda update-function-code --function-name samana_hello_dynamic \
16 --zip-file fileb://package_dynamic.zip
17
18
```

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": "Ndxc2A5z8XyLk9Fu359WeMXJ4o4Hp1q5Z9PqQMz0n1U=",
  "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": "Ndxc2A5z8XyLk9Fu359WeMXJ4o4Hp1q5Z9PqQMz0n1U="
```

```
{
  "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/3o0oYYLLDyxvdR6/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.

Lambda > Functions

Functions (36) Last fetched 10 seconds ago Actions Create function

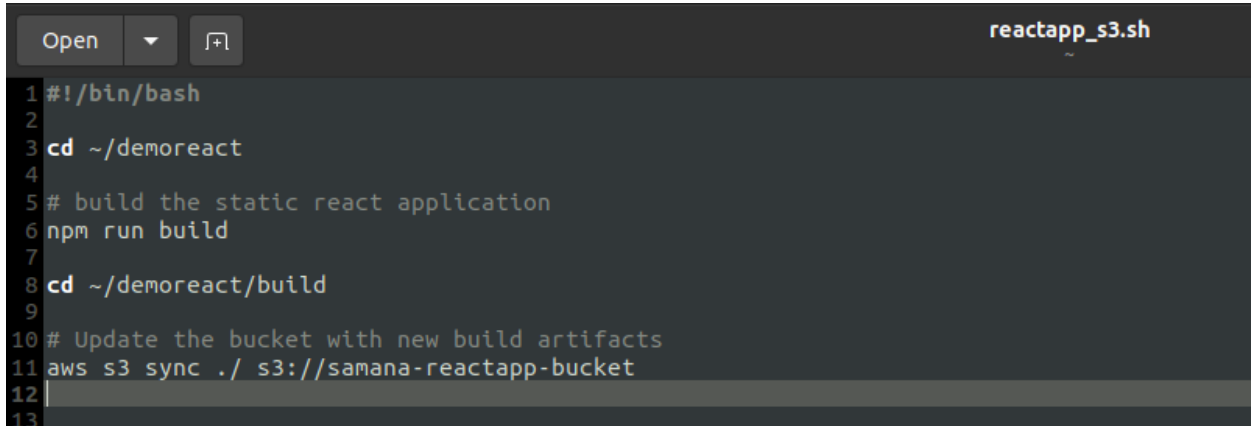
Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Function name	Description	Package type	Runtime	Code size	Last modified
<input type="checkbox"/>	samana_hello_dynamic	-	Zip	Python 3.9	330.0 byte	18 minutes ago
<input type="checkbox"/>	samana_lambda_hello	-	Zip	Python 3.9	300.0 byte	18 minutes ago

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

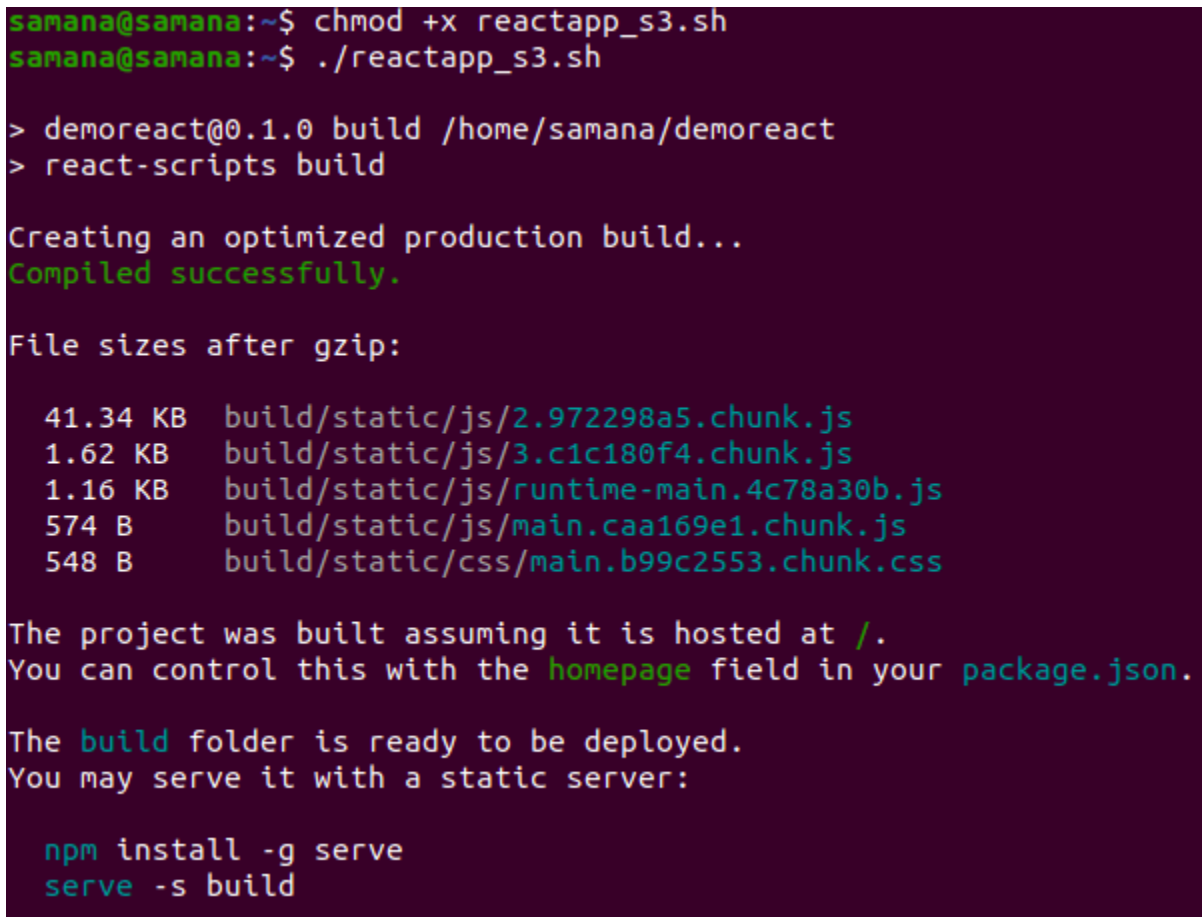
```
samana@samana:~$ ls demoreact/
node_modules package.json package-lock.json public README.md src
samana@samana:~$
```

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

A screenshot of a code editor window titled 'reactapp_s3.sh'. The editor has a dark theme and shows a shell script with line numbers 1 through 13. The script contains commands for building a React application and uploading it to an S3 bucket.

```
1#!/bin/bash
2
3cd ~/demoreact
4
5# build the static react application
6npm run build
7
8cd ~/demoreact/build
9
10# Update the bucket with new build artifacts
11aws s3 sync ./ s3://samana-reactapp-bucket
12
13
```

Running the script:

A screenshot of a terminal window showing the execution of the 'reactapp_s3.sh' script. The terminal output includes the command to make the script executable, the script's execution, the build process details, file sizes, and deployment instructions.

```
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:

samana-reactapp-bucket Info

Publicly accessible

[Objects](#)
[Properties](#)
[Permissions](#)
[Metrics](#)
[Management](#)
[Access Points](#)

Objects (8)

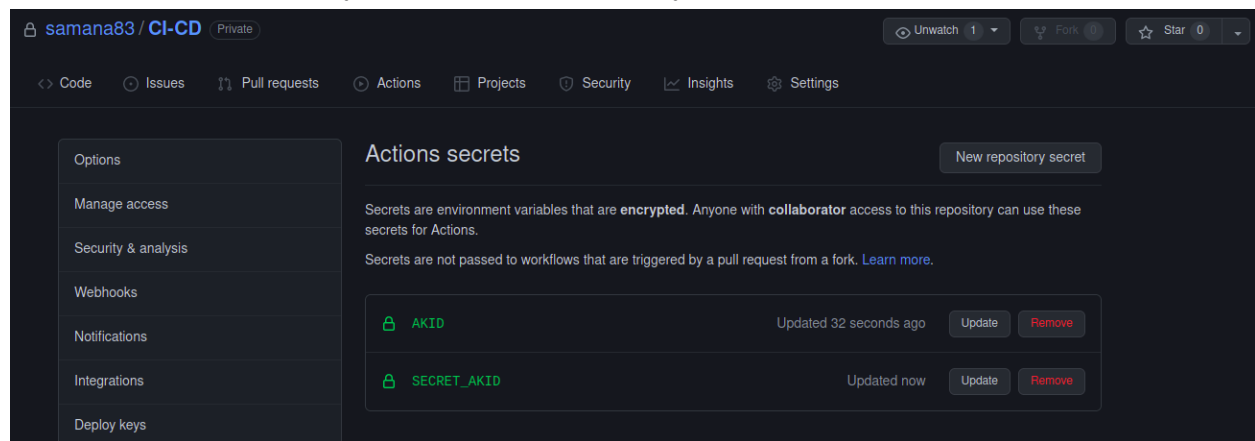
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<
1
>

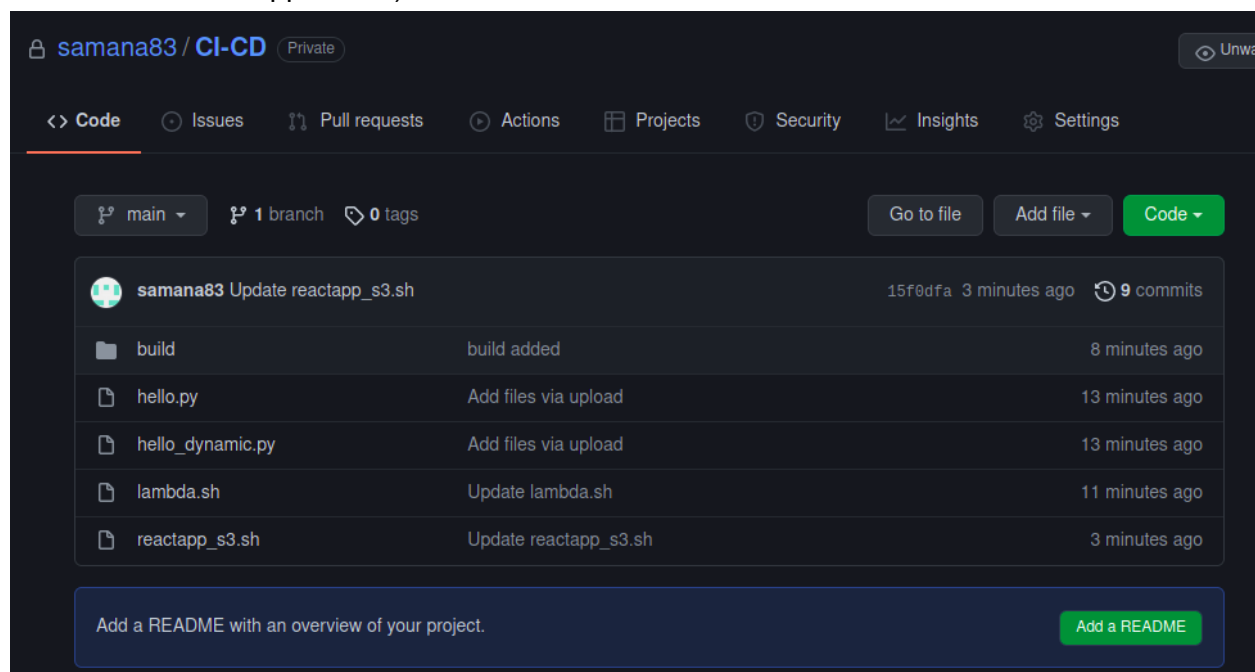
<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	asset-manifest.json	json	December 17, 2021, 17:02:02 (UTC+05:45)	1.1 KB	Standard
<input type="checkbox"/>	favicon.ico	ico	December 17, 2021, 17:02:01 (UTC+05:45)	3.8 KB	Standard
<input type="checkbox"/>	index.html	html	December 17, 2021, 17:02:02 (UTC+05:45)	3.0 KB	Standard
<input type="checkbox"/>	logo192.png	png	December 17, 2021, 17:02:02 (UTC+05:45)	5.2 KB	Standard
<input type="checkbox"/>	logo512.png	png	December 17, 2021, 17:02:02 (UTC+05:45)	9.4 KB	Standard

Now for integrating both of the above scripts we use Github actions

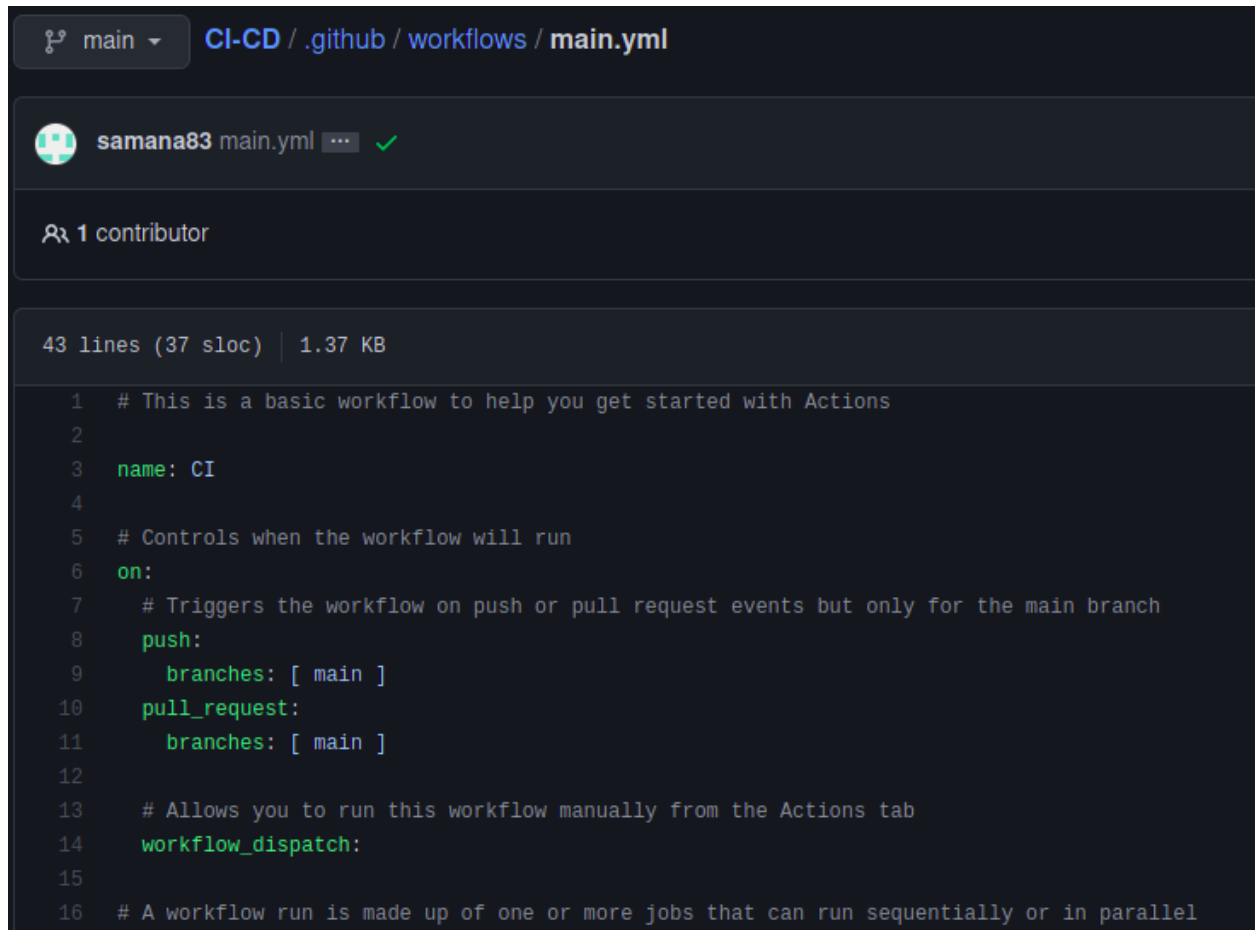
First we set the access key id and secret access key for our aws account as actions secret



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



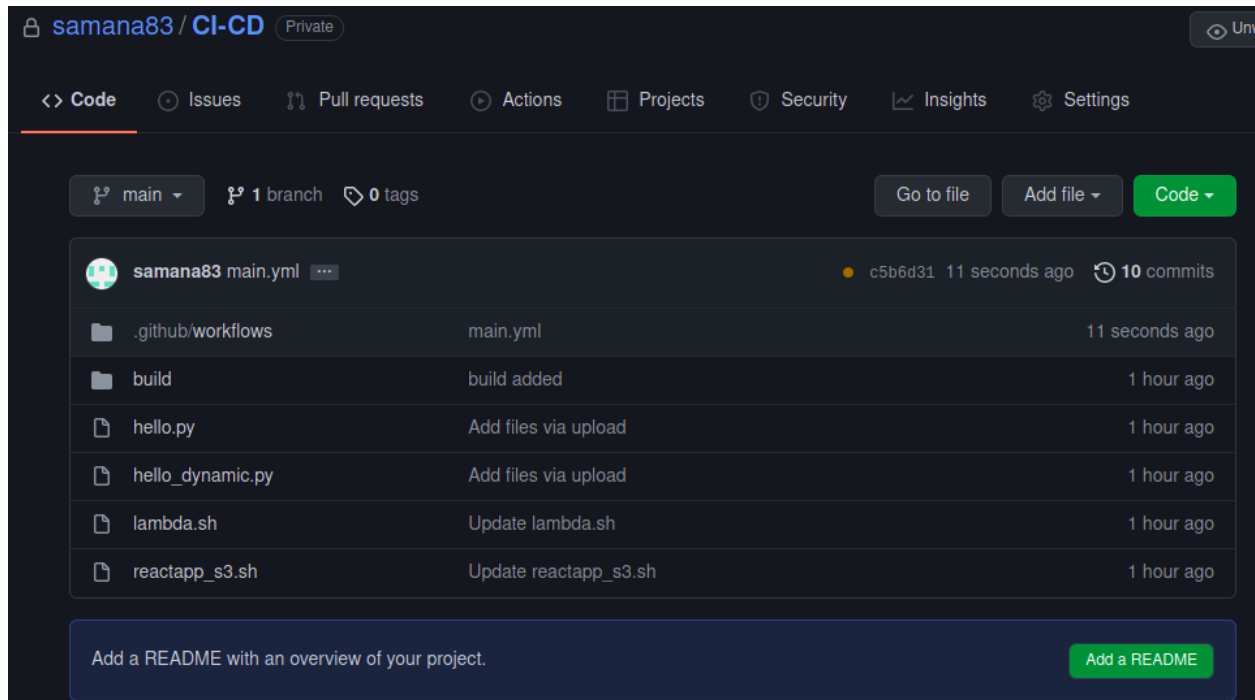
Now we create a main.yml workflow file as follows:



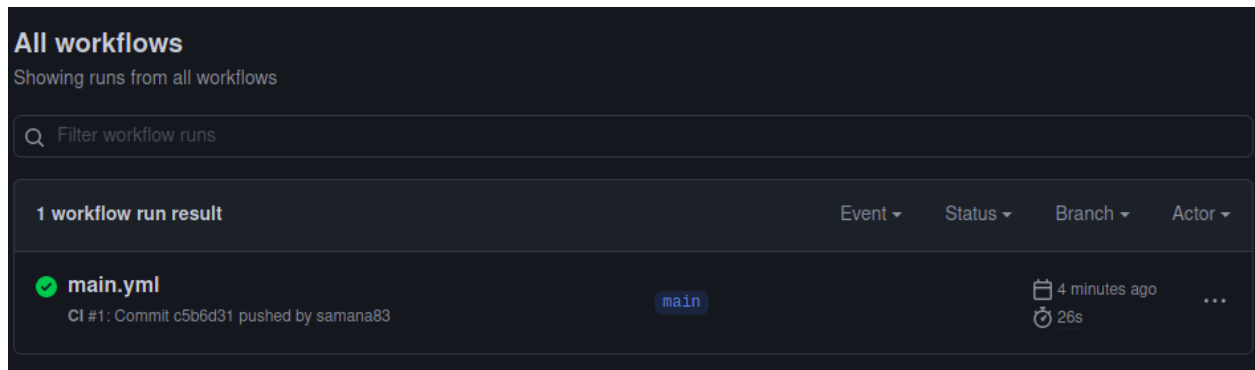
The screenshot shows the GitHub interface for a workflow file named `main.yml` located in the `CI-CD / .github / workflows` directory. The file is owned by `samana83` and has a green checkmark indicating it is valid. It has 1 contributor. The file size is 1.37 KB and it contains 43 lines (37 sloc). The workflow is named `CI` and is triggered on `push` and `pull_request` events for the `main` branch. It also includes a `workflow_dispatch` trigger for manual execution. The workflow is composed of one or more jobs that can run sequentially or in parallel.

```
1  # This is a basic workflow to help you get started with Actions
2
3  name: CI
4
5  # Controls when the workflow will run
6  on:
7    # Triggers the workflow on push or pull request events but only for the main branch
8    push:
9      branches: [ main ]
10   pull_request:
11     branches: [ main ]
12
13   # Allows you to run this workflow manually from the Actions tab
14   workflow_dispatch:
15
16   # A workflow run is made up of one or more jobs that can run sequentially or in parallel
```

```
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17 jobs:
18   # This workflow contains a single job called "build"
19   build:
20     # The type of runner that the job will run on
21     runs-on: ubuntu-latest
22
23     # Steps represent a sequence of tasks that will be executed as part of the job
24     steps:
25       # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26       - uses: actions/checkout@v2
27
28       # Runs a single command using the runners shell
29       - name: Configure AWS credentials
30         uses: aws-actions/configure-aws-credentials@v1
31         with:
32           aws-access-key-id: ${ secrets.AKID }
33           aws-secret-access-key: ${ secrets.secret_AKID }
34           aws-region: us-east-1
35
36       # Runs a set of commands using the runners shell
37       - name: Run the script for lambda functions
38         run: |
39           chmod +x lambda.sh
40           ./lambda.sh
41
42       - name: run the s3 reactapp script
43         run: |
44           chmod +x reactapp_s3.sh
45           ./reactapp_s3.sh
```

The workflow file was run:



The following build steps were successful

The screenshot shows the GitHub Actions interface for a workflow named 'main.yml' (CI #1). The build status is 'succeeded 5 minutes ago in 11s'. The left sidebar shows the 'build' job as successful. The main panel lists the steps of the 'build' job, all of which are successful:

Step	Duration
Set up job	4s
Run actions/checkout@v2	1s
Configure AWS credentials	1s
Run the script for lambda functions	3s
run the s3 reactapp script	2s
Post Configure AWS credentials	0s
Post Run actions/checkout@v2	0s
Complete job	0s

Configuring aws credentials:

The screenshot shows the 'Configure AWS credentials' step, which is successful. The step is defined in the workflow file as follows:

```
1  Run aws-actions/configure-aws-credentials@v1
2  with:
3    aws-access-key-id: ***
4    aws-secret-access-key: ***
5    aws-region: us-east-1
6
```

The script for lambda function was also run successfully.

✓ Run the script for lambda functions

```
1  ▶ Run chmod +x lambda.sh
10  adding: hello.py (deflated 18%)
11  adding: hello_dynamic.py (deflated 19%)
12  {
13    "FunctionName": "samana_lambda_hello",
14    "FunctionArn": "arn:aws:lambda:us-east-1:***:function:samana_lambda_hello",
15    "Runtime": "python3.9",
16    "Role": "arn:aws:iam:***:role/service-role/samana_role_hello",
17    "Handler": "lambda_function.lambda_handler",
18    "CodeSize": 300,
19    "Description": "",
20    "Timeout": 3,
21    "MemorySize": 128,
22    "LastModified": "2021-12-17T14:09:09.000+0000",
23    "CodeSha256": "rDW+UV5Ezkw5JgS04pa301vafj9afvqr0+16yUro54I=",
24    "Version": "$LATEST",
25    "TracingConfig": {
26      "Mode": "PassThrough"
27    },
28    "RevisionId": "48558755-17f5-4aad-a95c-fa2a5bd45ed3",
29    "State": "Active",
30    "LastUpdateStatus": "InProgress",
31    "LastUpdateStatusReason": "The function is being created.",
32    "LastUpdateStatusReasonCode": "Creating",
```

✓ Run the script for lambda functions

```
35     ]
36   }
37   {
38     "FunctionName": "samana_hello_dynamic",
39     "FunctionArn": "arn:aws:lambda:us-east-1:***:function:samana_hello_dynamic",
40     "Runtime": "python3.9",
41     "Role": "arn:aws:iam:***:role/service-role/samana_role_hello",
42     "Handler": "lambda_function.lambda_handler",
43     "CodeSize": 330,
44     "Description": "",
45     "Timeout": 3,
46     "MemorySize": 128,
47     "LastModified": "2021-12-17T14:09:10.000+0000",
48     "CodeSha256": "k+NE1s5N9/j5JylDYiVNd9ziSn208gUDFfAQ9cf2IGk=",
49     "Version": "$LATEST",
50     "TracingConfig": {
51       "Mode": "PassThrough"
52     },
53     "RevisionId": "d7ef6b5a-cb54-43d5-ae67-29089265e76f",
54     "State": "Active",
55     "LastUpdateStatus": "InProgress",
56     "LastUpdateStatusReason": "The function is being created.",
57     "LastUpdateStatusReasonCode": "Creating",
58     "Architectures": [
59       "x86_64"
60     ]
61   }
62 }
```

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

```
run the s3 reactapp script

1  ▶ Run chmod +x reactapp_s3.sh
10 Completed 3.8 KiB/583.0 KiB (20.2 KiB/s) with 19 file(s) remaining
11 upload: build/favicon.ico to s3://samana-reactapp-bucket/favicon.ico
12 Completed 3.8 KiB/583.0 KiB (20.2 KiB/s) with 18 file(s) remaining
13 Completed 4.7 KiB/583.0 KiB (13.7 KiB/s) with 18 file(s) remaining
14 upload: build/static/js/main.caa169e1.chunk.js to s3://samana-reactapp-bucket/static/js/main.caa169e1.chunk.js
15 Completed 4.7 KiB/583.0 KiB (13.7 KiB/s) with 17 file(s) remaining
16 Completed 7.8 KiB/583.0 KiB (18.2 KiB/s) with 17 file(s) remaining
17 upload: build/index.html to s3://samana-reactapp-bucket/index.html
18 Completed 7.8 KiB/583.0 KiB (18.2 KiB/s) with 16 file(s) remaining
19 Completed 8.8 KiB/583.0 KiB (20.1 KiB/s) with 16 file(s) remaining
20 upload: build/asset-manifest.json to s3://samana-reactapp-bucket/asset-manifest.json
21 Completed 8.8 KiB/583.0 KiB (20.1 KiB/s) with 15 file(s) remaining
22 Completed 8.9 KiB/583.0 KiB (19.0 KiB/s) with 15 file(s) remaining
23 upload: build/robots.txt to s3://samana-reactapp-bucket/robots.txt
24 Completed 8.9 KiB/583.0 KiB (19.0 KiB/s) with 14 file(s) remaining
25 Completed 14.1 KiB/583.0 KiB (29.9 KiB/s) with 14 file(s) remaining
26 upload: build/logo192.png to s3://samana-reactapp-bucket/logo192.png
27 Completed 14.1 KiB/583.0 KiB (29.9 KiB/s) with 13 file(s) remaining
28 Completed 15.6 KiB/583.0 KiB (32.9 KiB/s) with 13 file(s) remaining
29 upload: build/static/css/main.b99c2553.chunk.css.map to s3://samana-reactapp-bucket/static
/css/main.b99c2553.chunk.css.map
30 Completed 15.6 KiB/583.0 KiB (32.9 KiB/s) with 12 file(s) remaining
31 Completed 16.7 KiB/583.0 KiB (34.9 KiB/s) with 12 file(s) remaining
32 upload: build/static/css/main.b99c2553.chunk.css to s3://samana-reactapp-bucket/static/css/main.b99c2553.chunk.css
```

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

```
← → ↺ https://z2hphqr6xf.execute-api.us-east-1.amazonaws.com ☆
{"Hello": "Default"}
```

Result: Our react s3 application:

