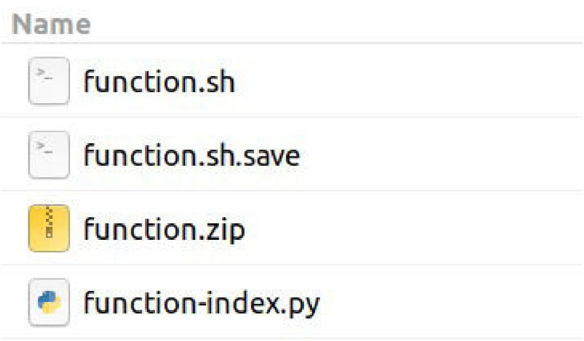
- Create a bash script to deploy your lambda functions
- Create a bash script to deploy your react app to S3
- Integrate both these scripts with one of Jenkins, Github Actions, CircleCI or TravisCI

1)
create a file of function-index.py

```
import json

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

```
saroj@saroj-Inspiron-3576:~/fucntion$ cat function.sh
#!/bin/bash
zip function.zip function-index.py
aws lambda create-function --function-name saroj-deploy --runtime python3.9 --zip-file fileb://function.zip --role arn:aws:iam::949263681218:role/service-role/saroj-apigateway1-role-lniyxd85 --handler function-index.lambda_handler --profile lft-training
```



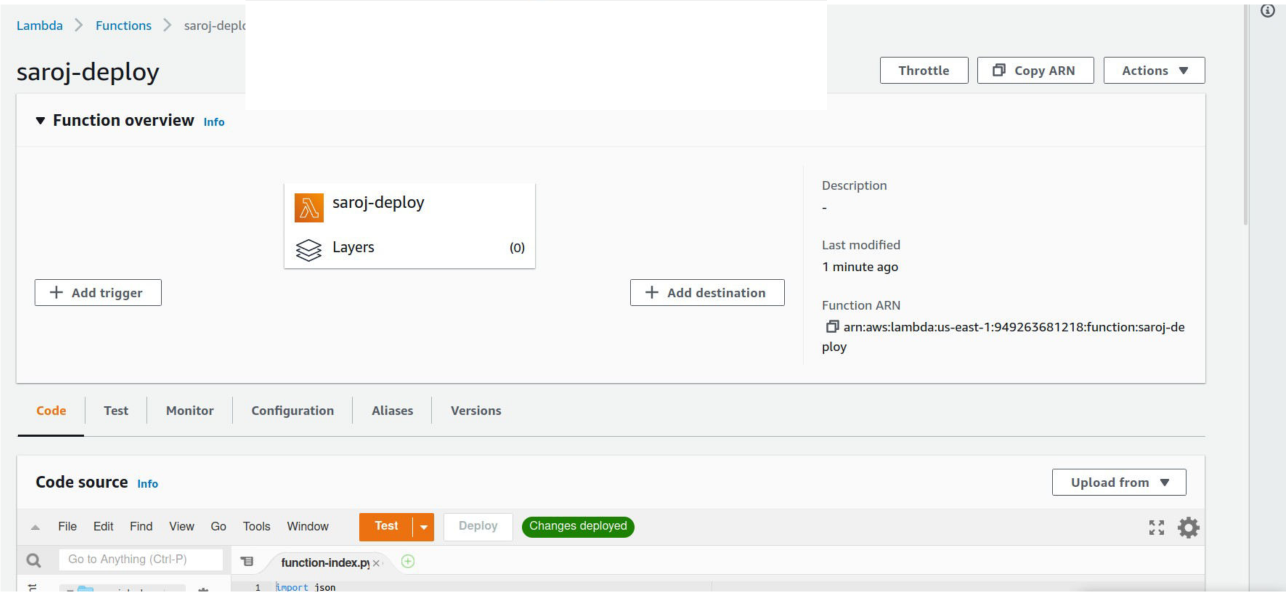
Name

function.sh

function.sh.save

function.zip

function-index.py



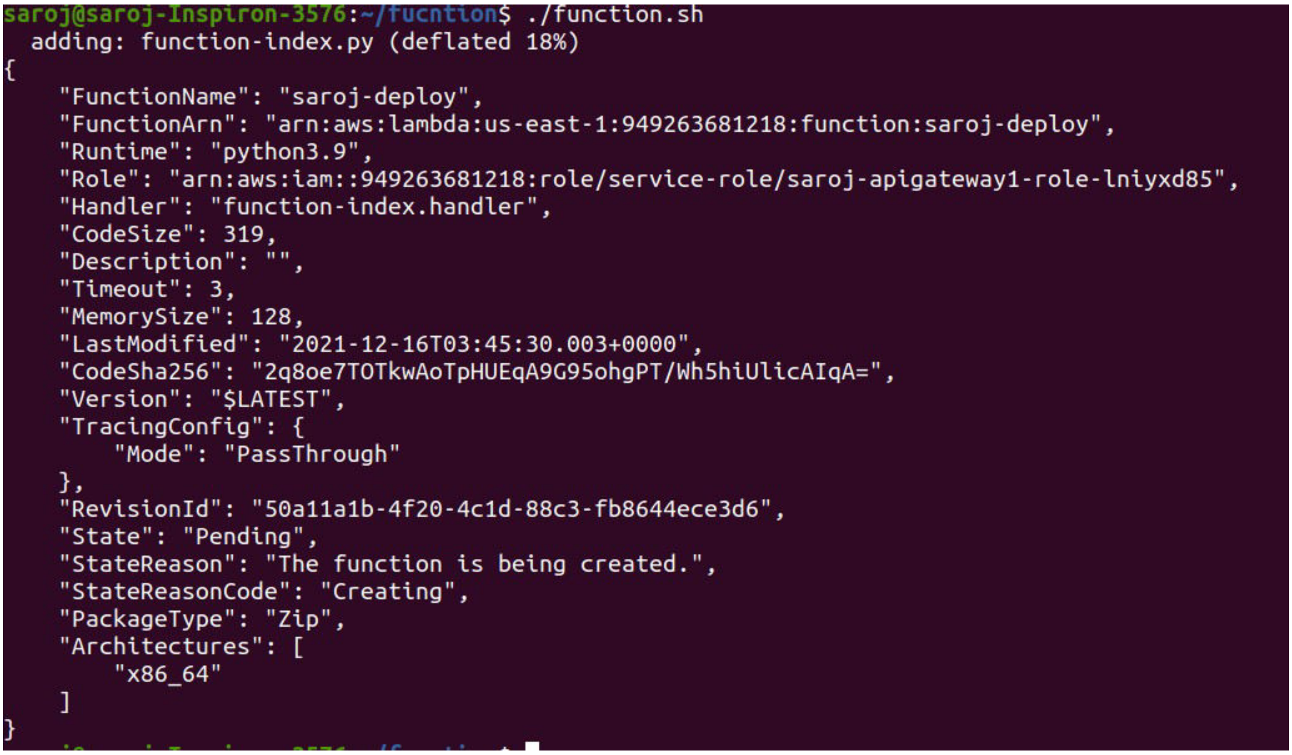
saroj-deploy

Description
-

Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:949263681218:function:saroj-deploy

```
import json
```

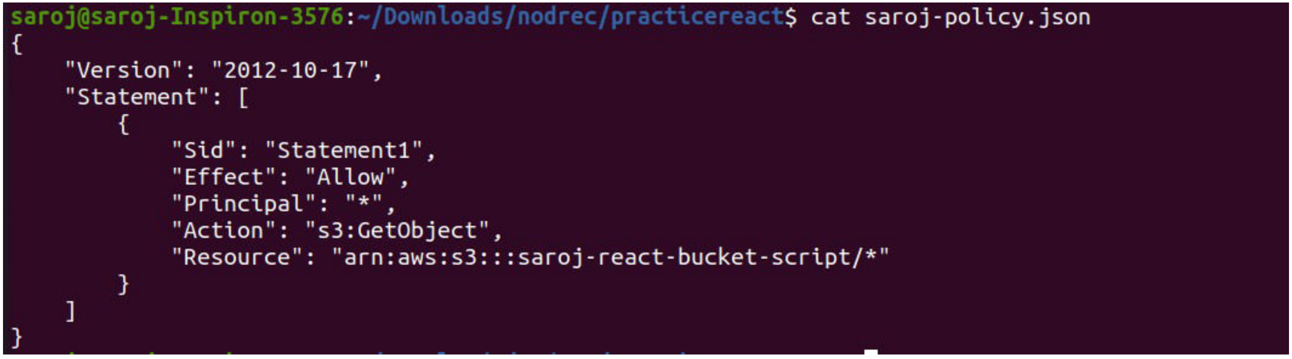


```
saroj@saroj-Inspiron-3576:~/fucntion$ ./function.sh
  adding: function-index.py (deflated 18%)
{
    "FunctionName": "saroj-deploy",
    "FunctionArn": "arn:aws:lambda:us-east-1:949263681218:function:saroj-deploy",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::949263681218:role/service-role/saroj-apigateway1-role-lniyxd85",
    "Handler": "function-index.handler",
    "CodeSize": 319,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2021-12-16T03:45:30.003+0000",
    "CodeSha256": "2q8oe7TOTkwAoTpHUEqA9G95ohgPT/Wh5hiUlicAIqA=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "50a11a1b-4f20-4c1d-88c3-fb8644ece3d6",
    "State": "Pending",
    "StateReason": "The function is being created.",
    "StateReasonCode": "Creating",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ]
}
```

2)

create a policy

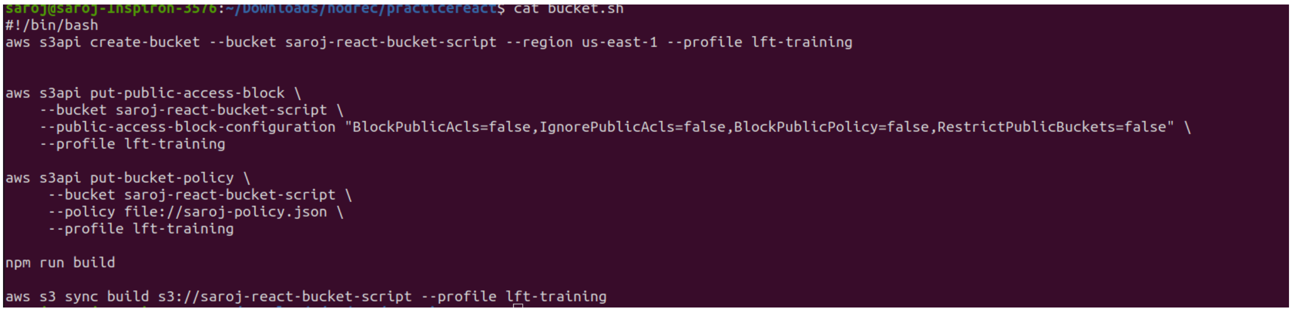
```
saroj@saroj-Inspiron-3576:~/Downloads/nodrec/practicereact$ cat saroj-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::saroj-react-bucket-script/*"
        }
    ]
}
```



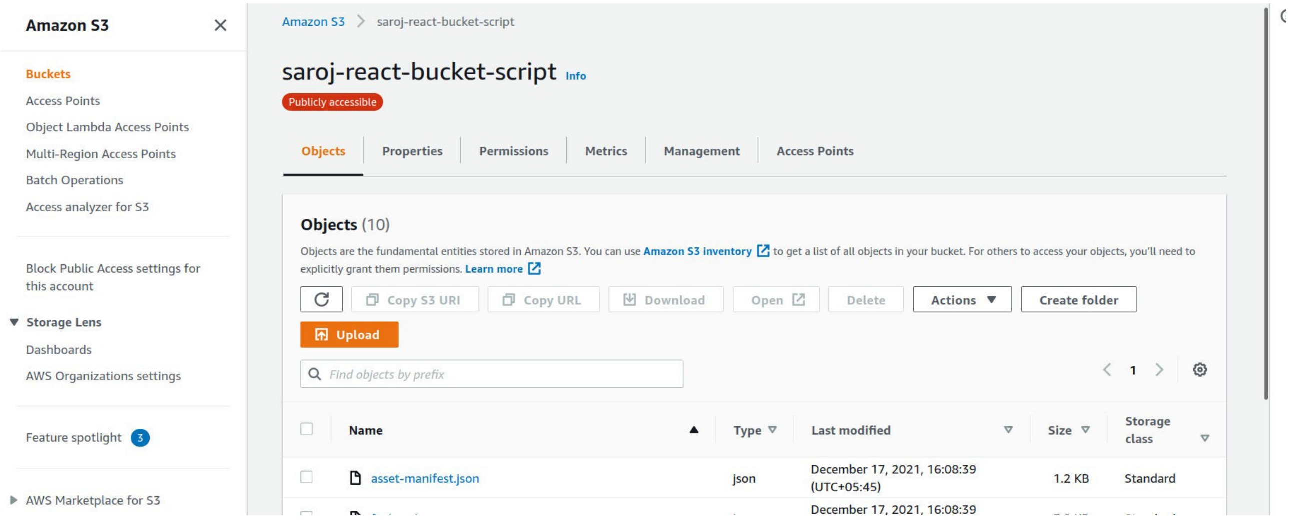
```
saroj@saroj-Inspiron-3576:~/Downloads/nodrec/practicereact$ cat saroj-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::saroj-react-bucket-script/*"
        }
    ]
}
```

script

```
#!/bin/bash
aws s3api create-bucket --bucket saroj-react-bucket-script --region us-east-1 --profile lft-training

aws s3api put-public-access-block \
    --bucket saroj-react-bucket-script \
    --public-access-block-configuration "BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false" \
    --profile lft-training

aws s3api put-bucket-policy \
     --bucket saroj-react-bucket-script \
     --policy file://saroj-policy.json \
     --profile lft-training

npm run build

aws s3 sync build s3://saroj-react-bucket-script --profile lft-training
```



```
saroj@saroj-Inspiron-3576:~/Downloads/nodrec/practicereact$ cat bucket.sh
#!/bin/bash
aws s3api create-bucket --bucket saroj-react-bucket-script --region us-east-1 --profile lft-training


aws s3api put-public-access-block \
    --bucket saroj-react-bucket-script \
    --public-access-block-configuration "BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false" \
    --profile lft-training

aws s3api put-bucket-policy \
     --bucket saroj-react-bucket-script \
     --policy file://saroj-policy.json \
     --profile lft-training

npm run build

aws s3 sync build s3://saroj-react-bucket-script --profile lft-training
```



Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > saroj-react-bucket-script

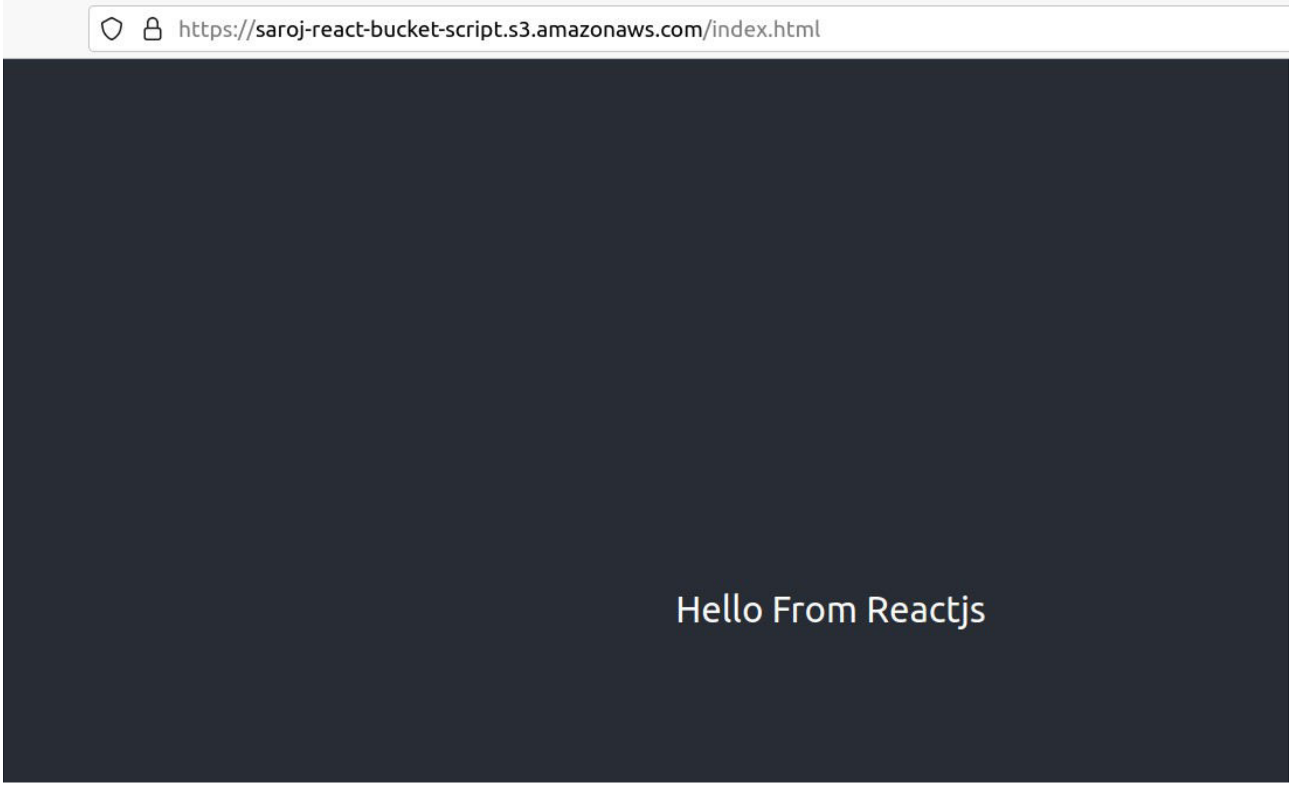
saroj-react-bucket-script Info

Publicly accessible

Objects (10)

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**

□	Name	Type	Last modified	Size	Storage class
□	asset-manifest.json	json	December 17, 2021, 16:08:39 (UTC+05:45)	1.2 KB	Standard
□			December 17, 2021, 16:08:39		

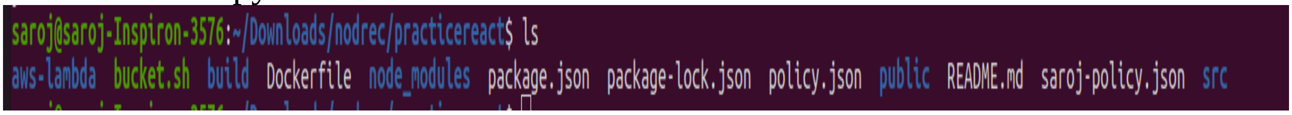


https://saroj-react-bucket-script.s3.amazonaws.com/index.html

Hello From Reactjs

3)

create a repo in git hub aws-lambda
upload a file
function-index.py for lambda and for the s3 react bucket below screen-shot files



```
saroj@saroj-Inspiron-3576:~/Downloads/nodrec/practicereact$ ls
aws-lambda  bucket.sh  build  Dockerfile  node_modules  package.json  package-lock.json  policy.json  public  README.md  saroj-policy.json  src
```

create a policy for the s3 named saroj-policy.json to make it public

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::saroj-react-bucket-script/*"
        }
    ]
}
```

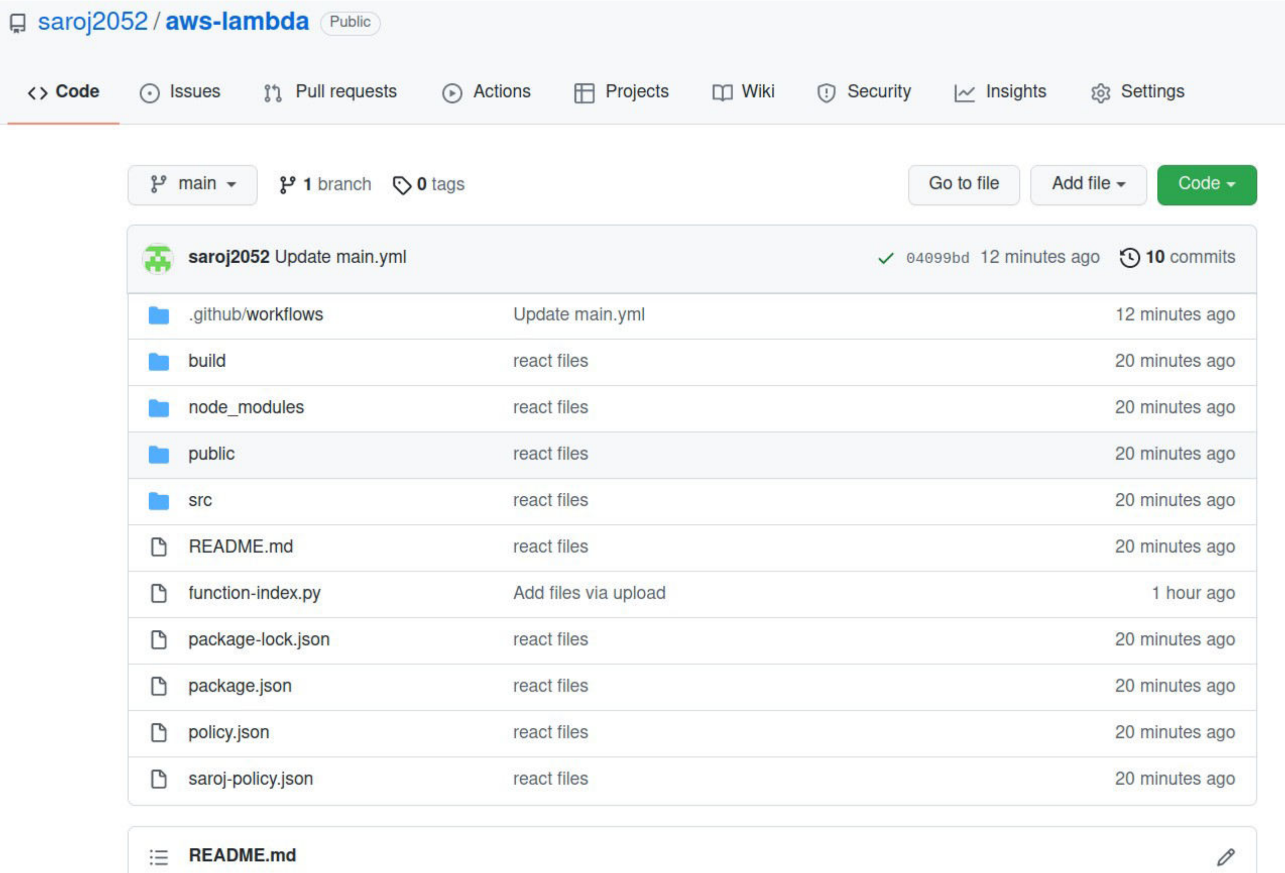
and then goto Actions
click on set a workfolw yourself

managed the secrets of aws

name the workflow
edit the script

```
name: CI

# 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

- 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-1
- name: creating Zip file of fuction-index.py
run: |
zip function.zip function-index.py

- name: Deploying lamda function to aws
run: |
aws lambda create-function --function-name saroj-deploy --runtime python3.9 \
--zip-file fileb://function.zip \
--role arn:aws:iam::949263681218:role/service-role/saroj-apigateway1-role-lniyxd85 \
--handler function-index.lambda_handler
```

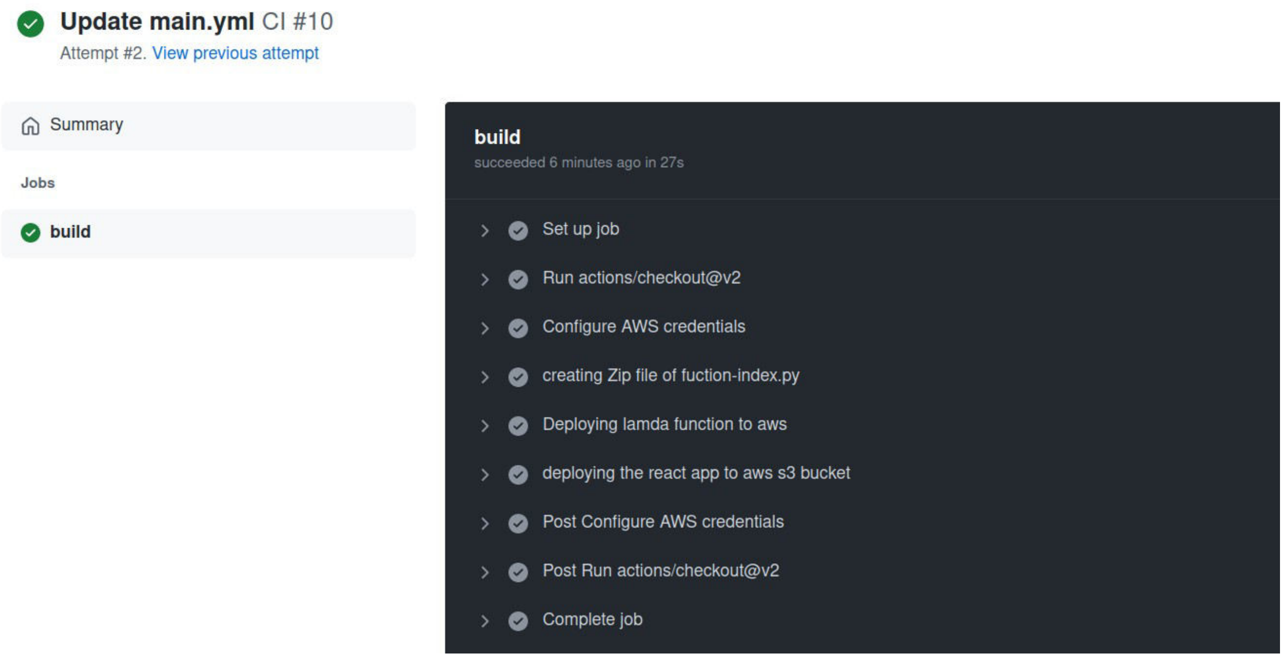
```
- name: deploying the react app to aws s3 bucket
run: |
aws s3api create-bucket --bucket saroj-react-bucket-script --region us-east-1

aws s3api put-public-access-block \
--bucket saroj-react-bucket-script \
--public-access-block-configuration
"BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false"

aws s3api put-bucket-policy \
--bucket saroj-react-bucket-script \
--policy file://saroj-policy.json

npm run build

aws s3 sync build s3://saroj-react-bucket-script
```



saroj2052 Update main.yml ✓ 04099bd 12 minutes ago 10 commits

.github/workflows	Update main.yml	12 minutes ago
build	react files	20 minutes ago
node_modules	react files	20 minutes ago
public	react files	20 minutes ago
src	react files	20 minutes ago
README.md	react files	20 minutes ago
function-index.py	Add files via upload	1 hour ago
package-lock.json	react files	20 minutes ago
package.json	react files	20 minutes ago
policy.json	react files	20 minutes ago
saroj-policy.json	react files	20 minutes ago

README.md



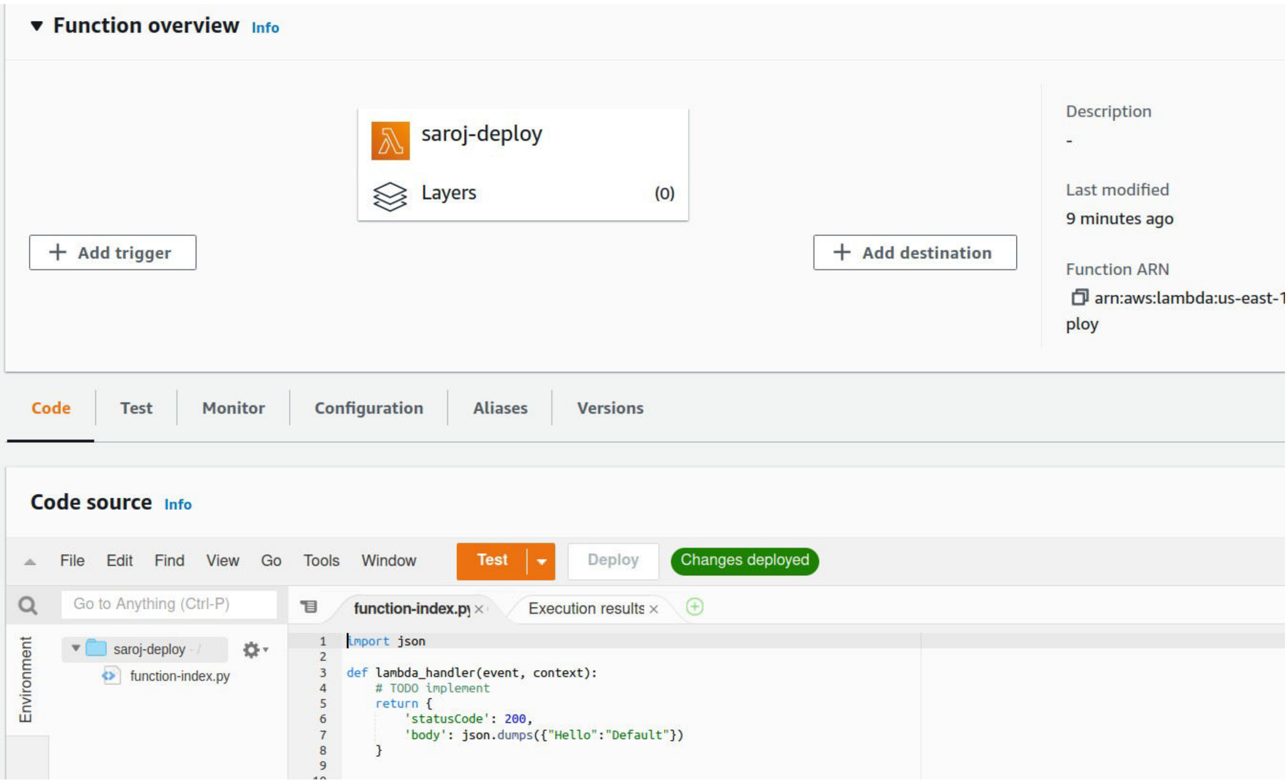
Update main.yml CI #10

Attempt #2. View previous attempt

build

succeeded 6 minutes ago in 27s

- Set up job
- Run actions/checkout@v2
- Configure AWS credentials
- creating Zip file of fuction-index.py
- Deploying lamda function to aws
- deploying the react app to aws s3 bucket
- Post Configure AWS credentials
- Post Run actions/checkout@v2
- Complete job



Function overview Info

saroj-deploy

Layers (0)

Description
-

Last modified
9 minutes ago

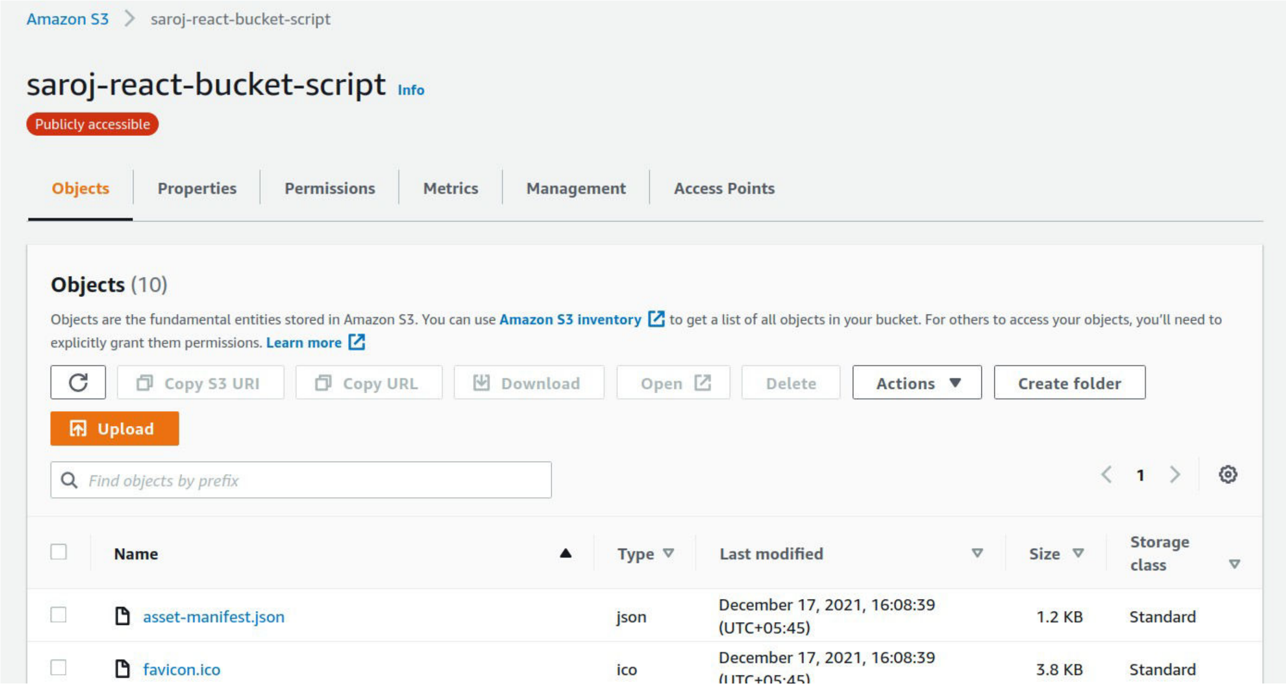
Function ARN
arn:aws:lambda:us-east-1
ploy

Code source Info

Changes deployed

function-index.py

```
import json

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

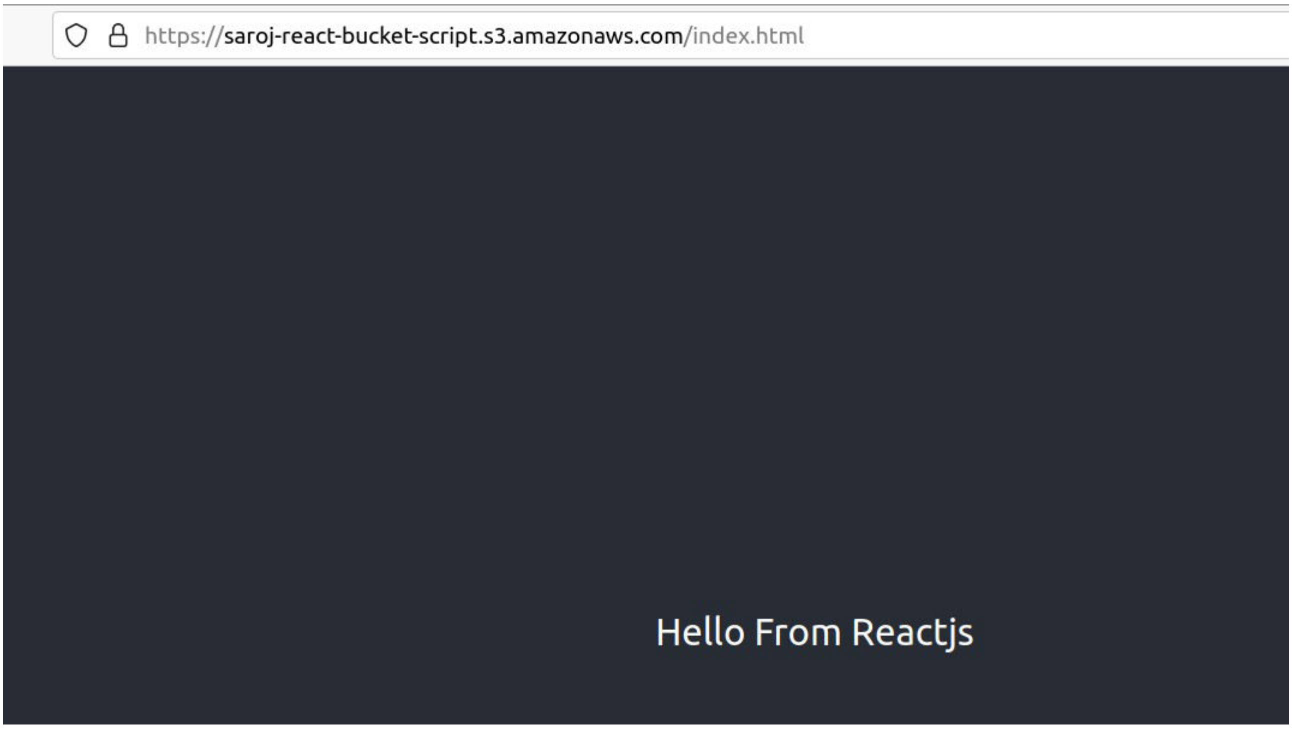


saroj-react-bucket-script

Objects (10)

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

□	Name	Type	Last modified	Size	Storage class
□	asset-manifest.json	json	December 17, 2021, 16:08:39 (UTC+05:45)	1.2 KB	Standard
□	favicon.ico	ico	December 17, 2021, 16:08:39 (UTC+05:45)	3.8 KB	Standard



https://saroj-react-bucket-script.s3.amazonaws.com/index.html

Hello From Reactjs

either we can also integrate the bash file to run section instead of writing commnads manually

I have done it manually because the –profile section 0f my bash file is showing the error because it I configured the aws section in secrets.