

Setup CI/CD Pipeline for React App using GitHub Actions

Before proceeding: make sure your application is currently being hosted on S3 using CloudFront, and your code has been moved to GitHub.

- Create a GitHub Action workflow within the folder where your app is located.

```
.github/workflows/build-and-deploy.yaml
```

- In this YAML file, configure all the required configurations as shown below.

```
---
name: Build and Deploy React App to CloudFront
on:
  push:
    branches: [ main ]
jobs:
  build-and-deploy:
    name: Build and Deploy
    runs-on: ubuntu-latest
    env:
      BUCKET: <bucket-name>
      DIST: build
      REGION: us-east-1
      DIST_ID: <ID>

    steps:
      - name: Checkout
        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 }
          aws-region: ${ env.REGION }

      # - uses: actions/setup-node@v2
      #   with:
      #     node-version: '14'
```

```

- name: Install Dependencies
  run: |
    node --version
    npm ci --production

- name: Build Static Website
  run: npm run build

- name: Copy files to the production website with the AWS CLI
  run: |
    aws s3 sync --delete ${ env.DIST } s3://${ env.BUCKET }

- name: Copy files to the production website with the AWS CLI
  run: |
    aws cloudfront create-invalidation \
      --distribution-id ${ env.DIST_ID } \
      --paths "/*"

```







- To allow GitHub Actions to upload files to S3 and create invalidation in CloudFront, it is necessary to generate a new IAM user and assign the necessary authorizations. We should make a policy like below.

```














{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "<your bucket arn>"
    },
    {
      "Sid": "AllObjectActions",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "<your bucket arn>/*"
    },
    {
      "Sid": "InvalidateCF",
      "Effect": "Allow",
      "Action": "cloudfront:CreateInvalidation",
      "Resource": "*"
    }
  ]
}

```

- Create a new user with the name "github-actions" and attach the policy specified above to it.
- Navigate to the GitHub repository and set up two secrets, namely `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`. These secrets will be used to authenticate and authorize GitHub actions that require access to your Amazon Web Services (AWS) account.

| Repository secrets | | |
|--|--------------------|---|
|  <code>AWS_ACCESS_KEY_ID</code> | Updated 1 hour ago |   |
|  <code>AWS_SECRET_ACCESS_KEY</code> | Updated 1 hour ago |   |

- The Continuous Integration/Continuous Deployment (CI/CD) pipeline has been successfully set up and you can verify its functionality by making changes to the source code. After committing and pushing the changes, the pipeline will be triggered, and you can observe the process in GitHub Actions.

| Build and Deploy | | |
|---|--|--|
| succeeded 48 minutes ago in 1m 2s | | <input type="text" value="Search logs"/>  |
| >  Set up job | | 3s |
| >  Checkout code | | 1s |
| >  Configure AWS Credentials | | 0s |
| >  Set up Node.js environment | | 0s |
| >  Install Dependencies | | 18s |
| >  Build Static Website | | 29s |
| >  Copy files to the production website with the AWS CLI | | 5s |
| >  Copy files to the production website with the AWS CLI | | 1s |
| >  Post Set up Node.js environment | | 0s |
| >  Post Configure AWS Credentials | | 0s |
| >  Post Checkout code | | 0s |
| >  Complete job | | 0s |

Author: Tewodroes Hailu
March 2023