

CI/CD Pipeline React AWS



i need create cicd pipeline for vite react project for upload the build artifact to the aws s3 bucket. and also connect the aws cloudfront to this s3 bucket also also using aws route 53.

To create a **CI/CD pipeline** for a **Vite + React** project that:

1. Builds the project using GitHub Actions
2. Uploads the build to **AWS S3**
3. Connects **CloudFront** to the S3 bucket
4. Uses **Route 53** for DNS

Here's a **complete guide** to achieve this setup:

✓ Step 1: Prerequisites

- An AWS S3 bucket (e.g. **my-react-app**)
- A CloudFront distribution (pointing to the S3 bucket)
- A Route 53 hosted zone with your domain (e.g. **example.com**)
- A valid SSL certificate in ACM (for HTTPS on CloudFront)
- IAM user/role with access to:
 - **s3:PutObject, s3>DeleteObject, s3:ListBucket**
 - **cloudfront:CreateInvalidation**
- GitHub repository set up with your Vite React project

✓ Step 2: Add GitHub Secrets

Go to your repository > **Settings** > **Secrets and variables** > **Actions** > **New repository secret**

Add:

Name	Value
AWS_ACCESS_KEY_ID	Your AWS access key
AWS_SECRET_ACCESS_KEY	Your AWS secret key
AWS_REGION	e.g., us-east-1
S3_BUCKET_NAME	e.g., my-react-app
CLOUDFRONT_DISTRIBUTION_ID	Your CloudFront Distribution ID

✓ Step 3: GitHub Actions Workflow (**.github/workflows/deploy.yml**)

yaml

```
name: Deploy Vite React App to S3 and CloudFront

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install dependencies
        run: npm ci

      - name: Build project
        run: npm run build

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v2
        with:
          aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
          aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
          aws-region: ${ secrets.AWS_REGION }

      - name: Sync files to S3
        run: |
          aws s3 sync dist/ s3://${ secrets.S3_BUCKET_NAME } \
            --delete \
            --cache-control "no-cache" \
            --acl public-read

      - name: Invalidate CloudFront cache
        run: |
          aws cloudfront create-invalidation \
            --distribution-id ${ secrets.CLOUDFRONT_DISTRIBUTION_ID } \
            --paths "/*"
```

✓ Step 4: Configure vite.config.js (optional for routing)

If you're using React Router:

```
js

// vite.config.js
export default defineConfig({
  plugins: [react()],
  build: {
    outDir: 'dist',
  },
  base: '/',
});
```

Ensure your S3 bucket has:

- Static website hosting enabled

- Index document: `index.html`
- Error document: `index.html` (for SPA routing)

✓ Step 5: S3 Bucket Configuration

1. Enable **Static website hosting**
2. Set **public read permissions** or use CloudFront + OAI (for private bucket)
3. Set bucket policy if needed:

```
json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-react-app/*"
    }
  ]
}
```

✓ Step 6: Route 53 Setup

1. Go to **Route 53** > Hosted zones > your domain
2. Create a **record set**:
 - **Type**: A – IPv4 address
 - **Alias**: Yes
 - **Alias Target**: CloudFront distribution

✓ Bonus: Custom Domain with HTTPS

1. Issue an SSL certificate in **ACM** (in `us-east-1`)
2. Attach it to your **CloudFront Distribution**
3. Set **Alternate domain names (CNAMEs)** to your custom domain (e.g., `www.example.com`)

Would you like me to:

- ✓ Generate the `vite.config.js` and `package.json` scripts
- ✓ Help set up a CloudFront distribution with OAI for better security
- ✓ Provide a sample DNS/ACM/CloudFront Terraform or CloudFormation script

Let me know what you need next!