

# AWS CI/CD Pipeline for Static Website

This project demonstrates a full CI/CD pipeline setup to automatically deploy a static website hosted on **Amazon S3** using:

- **GitHub** for source code
- **AWS CodePipeline** for orchestration
- **AWS CodeBuild** for build automation
- **Amazon S3** for static website hosting

Project Structure (GitHub Repository):

```
aws-cicd-s3/
    ├── index.html
    ├── about.html
    └── buildspec.yml
```

## **Index.html**

```
<!DOCTYPE html>

<html>
  <head>
    <title>My Static Site</title>
  </head>
  <body>
    <h1>Welcome to My Static Site</h1>
    <p>This website is deployed via AWS CI/CD pipeline automatically!</p>
  </body>
</html>
```

## **About.html**

```
<!DOCTYPE html>

<html>
  <head>
    <title>About</title>
  </head>
  <body>
    <h1>About Us</h1>
    <p>This is a static site using CodePipeline.</p>
  </body>
</html>
```

## **buildspec.yml**

```
version: 0.2
```

```
phases:
```

```
  install:
```

```
    commands:
```

```
      - echo Installing...
```

```
  build:
```

```
    commands:
```

```
      - echo Build started on `date`
```

```
      - echo Building the static site...
```

```
artifacts:
```

```
  files:
```

```
    - '**/*'
```

## Step 1: Create a GitHub Repository

1. Create a new repository on GitHub: aws-cicd-s3
2. Add your static site files: index.html, about.html, and buildspec.yml
3. Initialize git and push your files:

```
git init  
git remote add origin https://github.com/your-username/aws-cicd-s3.git  
git add .  
git commit -m "Initial commit"  
git branch -M main  
git push -u origin main
```

## Step 2: Create an S3 Bucket for Hosting

- ◆ 1. Go to <https://s3.console.aws.amazon.com/>

Click **Create bucket**

- ◆ 2. Bucket Settings:

Option	Value/Action
Bucket name	raju-cicd-site-01 ( <i>must be unique</i> )
Region	(Choose your nearest AWS region)
Block public access	 Uncheck all, then confirm
Versioning	Optional (can leave OFF)
Default settings	Keep all other default values

Then click **Create Bucket**

The screenshot shows the AWS S3 Buckets page. At the top, there's a search bar and navigation links for 'Amazon S3 > Buckets'. A banner at the top says 'Account snapshot - updated every 24 hours' with a link to 'All AWS Regions'. Below the banner, it says 'Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more' with a link to 'View Storage Lens dashboard'. There are tabs for 'General purpose buckets' (which is selected) and 'Directory buckets'. A button for 'Create bucket' is visible. The main table lists one bucket:

Name	AWS Region	IAM Access Analyzer	Creation date
raju-cicd-site-01	Asia Pacific (Mumbai) ap-south-1	<a href="#">View analyzer for ap-south-1</a>	July 8, 2025, 18:00:39 (UTC+05:30)

At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

### ◆ 3. Enable Static Website Hosting:

1. Click into your new bucket raju-cicd-site-01
2. Go to the **Properties** tab
3. Scroll to **Static website hosting**
4. Click **Edit**:
  - Enable:  **Enable**
  - Index document: index.html
  - Error document: (optional) 404.html
5. Click **Save changes**

◆ **4. Make Bucket Public (for website access):**

1. Go to **Permissions > Bucket Policy**
2. Click **Edit**
3. Paste this policy (replace bucket name if needed):

json

CopyEdit

{

```
"Version":"2012-10-17",
"Statement": [
    {
        "Sid":"PublicReadGetObject",
        "Effect":"Allow",
        "Principal":"*",
        "Action":["s3:GetObject"],
        "Resource":["arn:aws:s3:::raju-cicd-site-01/*"]
    }
]
```

4. Click **Save changes**

◆ **5. Note the Website URL:**

Go back to **Properties > Static Website Hosting**, and note the **Website Endpoint**. It will look like:

<http://raju-cicd-site-01.s3-website-us-east-1.amazonaws.com>

This will show your website once CodePipeline deploys it.

### **Step 3: Create a CodeBuild Project**

1. ◆ **1. Click Create Build Project**

2. ◆ **2. Project Configuration**

Field	Value
<b>Project name</b>	cicd-s3-build
<b>Description</b>	(optional) Build & deploy static site

◆ **3. Source Section**

Field	Value
<b>Source Provider</b>	GitHub
<b>Repository</b>	Connect your GitHub account
<b>Repo</b>	Raju6615/aws-cicd-s3
<b>Branch</b>	main

◆ **4. Environment Section**

Field	Value
<b>Environment image</b>	Managed image
<b>Operating system</b>	Ubuntu
<b>Runtime(s)</b>	Standard

Field	Value
<b>Image version</b>	Always use the latest
<b>Service role</b>	Select existing → Choose codebuild-s3-role

## ◆ 5. Buildspec

Select:  Use a buildspec file

File name: buildspec.yml (it's already in your repo)

## ◆ 6. Artifacts

Field	Value
<b>Type</b>	Amazon S3
<b>Bucket name</b>	raju-cicd-site-01 (select from list)
<b>Name</b>	(optional, e.g.,) output/
<b>Packaging</b>	None
<b>Encryption</b>	Disabled (for public static site)

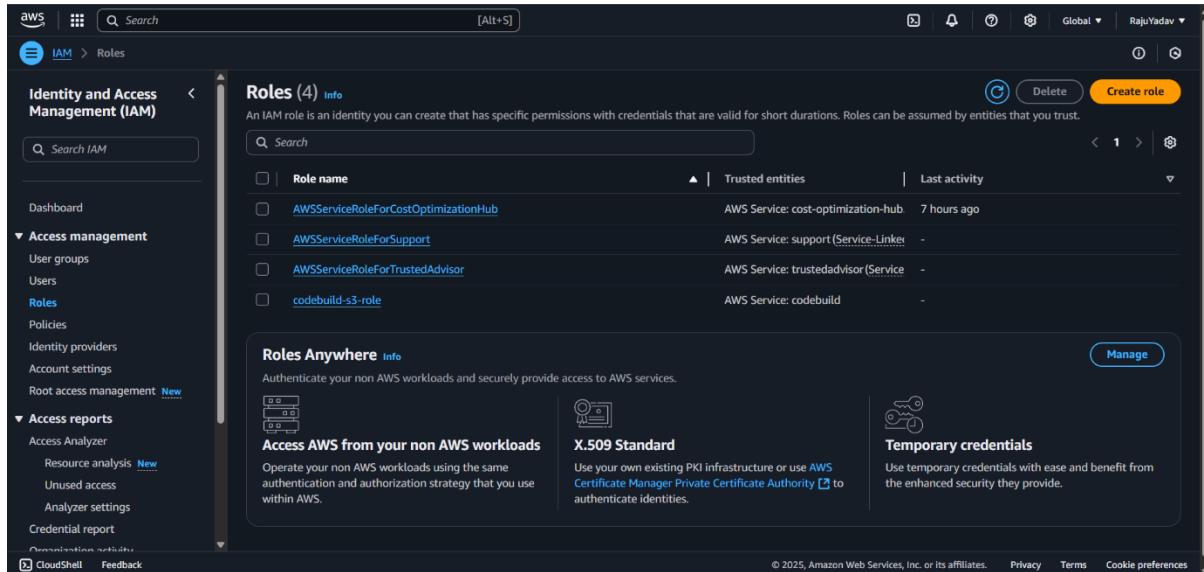
Then click **Create build project**

## Create IAM Role for CodeBuild :

### 1. Go to IAM > Roles > Create Role

#### ◆ 2. Trusted Entity Type:

- Select **AWS service**
- Use case: **CodeBuild**
- Click **Next**



#### ◆ 3. Attach Permissions Policies

On the **permissions screen**, search and select these two policies:

##### Policy Name

AmazonS3FullAccess

##### Why Needed?

So CodeBuild can upload to your S3 site



CloudWatchLogsFullAccess

So it can log build status/output

Then click **Next**

#### ◆ 4. Name the Role

- Role name: codebuild-s3-role

- Description: *(Optional)* Role for CodeBuild to deploy to S3
- Click **Create Role**

 Once done, your IAM role is ready to be used in the CodeBuild project.

#### Step 4: Create a CodePipeline

##### 1. Click Create pipeline

**Field**                   **Value**

**Pipeline name**       static-site-deploy

**Service role**          Choose: New service role (default is fine)

**Advanced settings** Leave default

 Click Next

##### ◆ 2. Add Source Stage

**Field**                   **Value**

**Source provider**       GitHub (Version 2)

**Connection**           Select github-raju-connection

**Repository**           Raju6615/aws-cicd-s3

**Branch name**          main

**Change detection**     Amazon Webhook

 Click Next

Raju6615 / aws-cicd-s3

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Code

main 1 Branch 0 Tags Go to file Add file < Code

Raju6615 Merge branch 'main' of https://github.com/Raju6615/aws-cicd-s3 6309364 · 1 hour ago 3 Commits

README.md Initial commit 1 hour ago

about.html Initial CI/CD site files 1 hour ago

buildspec.yml Initial CI/CD site files 1 hour ago

index.html Initial CI/CD site files 1 hour ago

README

aws-cicd-s3

About No description, website, or topics provided.

Readme Activity 0 stars 0 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages

### ◆ 3. Add Build Stage

Field Value

Build provider AWS CodeBuild

Project name cicd-s3-build

Click Next

ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/static-site-deploy/view?region=ap-south-1

Success The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Developer Tools > CodePipeline > Pipelines > static-site-deploy

static-site-deploy Edit Stop execution Create trigger Clone pipeline Release change

Pipeline Executions Triggers Settings Tags Stage

Source Build Deploy

All actions succeeded.

Source: Updated by GitHub Just now

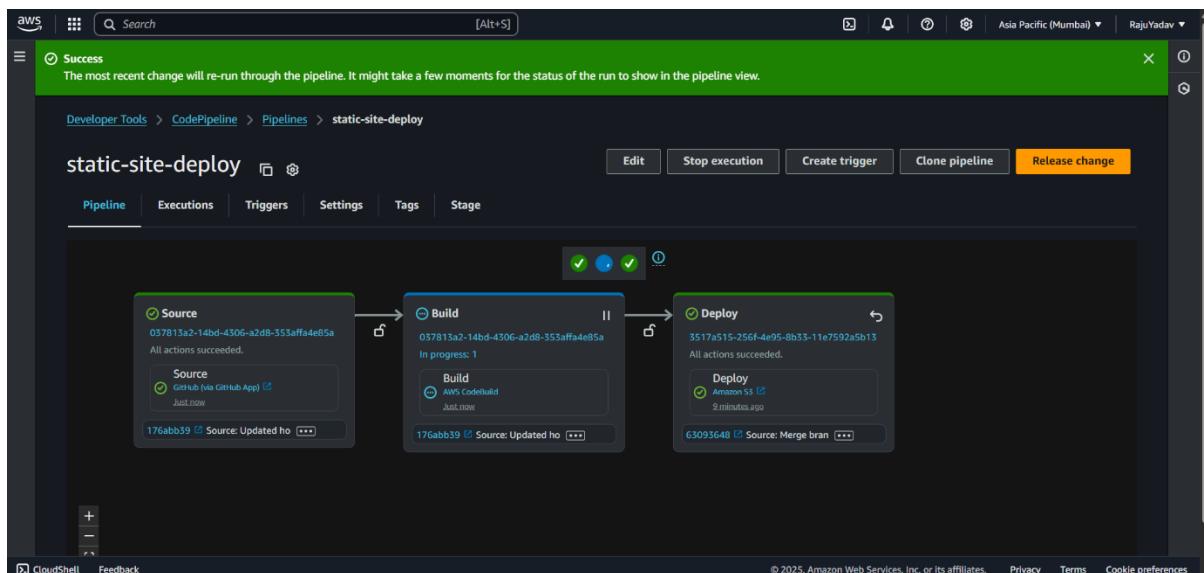
Build: AWS CodeBuild Just now

Deploy: Amazon S3 Just now

#### ◆ 4. Add Deploy Stage

Field	Value
Deploy provider	<input checked="" type="checkbox"/> Amazon S3
Region	Your selected region
Bucket	raju-cicd-site-01
Extract file	<input checked="" type="checkbox"/> Check this option (Yes)

Click **Next**, then **Create Pipeline**



The screenshot shows the AWS CodePipeline console with a green header bar indicating a 'Success' status. The main area displays a pipeline named 'static-site-deploy' with three stages: Source, Build, and Deploy. Each stage has a green checkmark icon and a success message. The Source stage shows a GitHub trigger. The Build stage uses AWS CodeBuild. The Deploy stage uses Amazon S3. The pipeline is currently at the 'All actions succeeded' state.

## Test the Automation

1. Edit index.html locally (change heading or paragraph)
2. Run:

```
git add .
```

```
git commit -m "Updated homepage"
```

```
git push
```

3. Wait ~1–2 mins. Check the website — it will be auto-updated.





## ■ Conclusion

In this project, we successfully implemented a **fully automated CI/CD pipeline** using **AWS Developer Tools** to deploy a static website hosted on **Amazon S3**. By integrating **GitHub**, **AWS CodePipeline**, and **AWS CodeBuild**, we achieved an efficient and serverless deployment workflow that:

- Automatically pulls source code from GitHub
- Builds and packages the site using a buildspec.yml
- Deploys the output directly to an S3 bucket configured for static website hosting

This project demonstrates how cloud-native DevOps practices can be applied to real-world scenarios with **zero manual intervention** after the initial setup. The pipeline is **scalable, cost-effective**, and follows best practices for deploying static web content.

Additionally, we ensured the setup was cost-optimized by cleaning up unused resources such as the CodeBuild project and pipeline after successful deployment. The GitHub connection remains in place for potential future reuse.

- ✓ This project lays a strong foundation for deploying modern websites and applications on AWS using continuous integration and delivery principles.