

Project Documentation: AWS Static Website CI/CD Pipeline

1. Project Setup Overview

Goal: Build a CI/CD pipeline to deploy a static website (HTML, CSS, JS) on AWS.

Key AWS services used: S3, CodePipeline, CodeBuild, IAM, CloudFront, CloudWatch, SNS.

GitHub repository as the source of pipeline.

2. Prepare the Source Repository

Set up a GitHub repository to store the static website files (index.html and images). This repository will serve as the source for the CI/CD pipeline.

[Create a new repository](#)

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * lizh1994 / Repository name * aws-static-site-cicd-lizhu
✓ aws-static-site-cicd-lizhu is available.

Great repository names are short and memorable. How about [curly-couscous](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Add README On

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore No .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

Add license MIT License

Licenses explain how others can use your code. [About licenses](#)

[Create repository](#)

Add initial static website files (index.html, images), commit and push changes.

3. Create and Configure S3 Buckets

Create an S3 bucket for static website hosting.

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the [AWS Simple Monthly Calculator](#)

View pricing details

How it works

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express interface for faster processing of data within a single Availability Zone.

Bucket name Info

lizhu-static-site-demo-s3

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Configuration:

- Block Public Access: unchecked (required for static hosting).
- Versioning: enabled.

- Encryption: enabled with SSE-S3.

- Bucket Policy: restricted to s3:GetObject only to allow CloudFront access.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

- Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

▶ Advanced settings

ⓘ After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#)

[Create bucket](#)

General purpose buckets [All AWS Regions](#)

Directory buckets

General purpose buckets (1) [Info](#)

Buckets are containers for data stored in S3.



[Copy ARN](#)

[Empty](#)

[Delete](#)

[Create bucket](#)

ⓘ Find buckets by name

< 1 > |

Name	AWS Region	Creation date
lizhu-static-site-demo-s3	US East (N. Virginia) us-east-1	September 27, 2025, 16:17:44 (UTC+01:00)

Amazon S3 > Buckets > lizhu-static-site-demo-s3

ⓘ Successfully edited bucket policy.

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Edit](#) [Delete](#)

[Copy](#)

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

Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Disable
 Enable

Hosting type

Host a static website
 Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
 Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

error.html

Upload the initial index.html and images, and test the static site via the S3 object URL.

Files and folders (2 total, 3.5 MB)

All files and folders in this table will be uploaded.

Find by name					Remove	Add files	Add folder
checkbox	Name	Type	Type	Size			
<input checked="" type="checkbox"/>	index.html	-	text/html	1.5 KB			
<input checked="" type="checkbox"/>	profile.jpg	-	image/jpeg	3.5 MB			

Destination Info

Destination
[s3://lizhu-static-site-demo-s3](#)

Destination details

Bucket settings that impact new objects stored in the specified destination.

Permissions

Grant public access and access to other AWS accounts.

ⓘ This bucket has the [bucket owner enforced](#) setting applied for Object Ownership. Use bucket policies to control access. [Learn more](#)

lizhu-static-site-demo-s3 Info

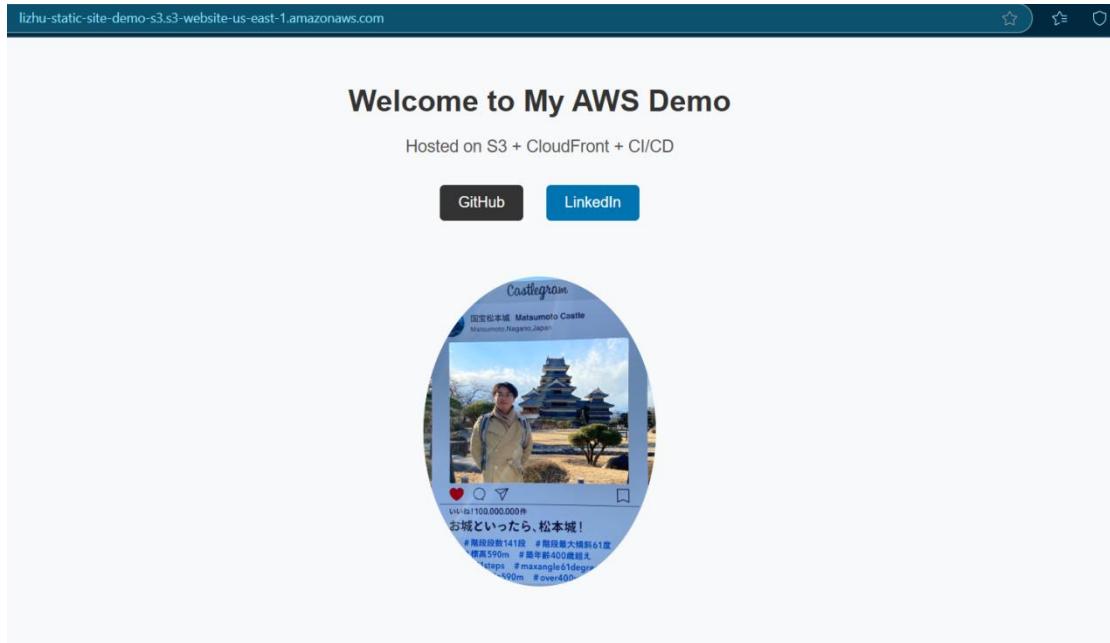
Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Objects (2)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

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](#)

Find objects by prefix							Show versions			
checkbox	Name	Type	Last modified	Size	Storage class					
<input type="checkbox"/>	index.html	html	September 27, 2025, 16:33:43 (UTC+01:00)	1.5 KB	Standard					
<input type="checkbox"/>	profile.jpg	jpg	September 27, 2025, 16:33:46 (UTC+01:00)	3.5 MB	Standard					



In this project, the deploy S3 bucket has a bucket policy that grants public read access (s3:GetObject) for website visitors.

At the same time, the CodeBuild IAM Role is attached with permissions to read/write/delete/list objects in the deploy bucket.

Key Point:

- Bucket Policy controls who (principals such as public, other accounts) can access the bucket.
- IAM Role Policy controls what the CodeBuild service itself can do.
- The final permission = intersection of both (the request must be allowed by both IAM + Bucket policy).

Why this matters:

This design ensures public visitors only have read access, while CodeBuild has full update capability. It's a clear example of IAM + Bucket Policy collaboration, aligning with the principle of least privilege.

4. Configure CloudFront Distribution and Functions

Create a CloudFront distribution with the Deploy Bucket as the origin.

Amazon CloudFront

Securely deliver content with low latency and high transfer speeds

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.

Get started with CloudFront

Enable accelerated, reliable and secure content delivery for Amazon S3 buckets, Application Load Balancers, Amazon API Gateway APIs, and more in 5 minutes or less.

[Create a CloudFront distribution](#)

Benefits and features

Reduce latency

The CloudFront network has 225+ points of presence (PoPs) that are connected by fully redundant, parallel 100 GbE fiber delivering ultra-low latency performance.

Improve security

Use CloudFront for perimeter protection, traffic encryption, and access controls. AWS Shield Standard defends traffic transmitted through CloudFront from

AWS Free Tier

1 TB of data transfer out

10,000,000 HTTP or HTTPS requests

2,000,000 CloudFront Function invocations

Each month, always free

CloudFront > Distributions > Create distribution

Get started

Step 2
Specify origin
Step 3
Enable security
Step 4
Get TLS certificate
Step 5
Review and create

Get started

Connect your websites, apps, files, video streams, and other content to CloudFront. We optimize the performance, reliability, and security for your web traffic.

Distribution options

Distribution name
Name will be stored as a tag on the resource. You can add a name, or more tags, later.

Description - optional

Distribution type

Single website or app
Choose if each website or application will have a unique configuration.

Multi-tenant architecture - New
Choose when you have multiple domains that need to share configurations. This is a common architecture for SaaS providers.

Custom domain

Domain - optional
Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app. You can add a custom domain later if you do not have a Route 53 zone in this account.

Tags - optional

Specify origin

Origin type

Your origin is where your content (such as a website or app) lives. CloudFront works with AWS-based origins and origins hosted on other cloud providers.

Amazon S3

Deliver static assets like files and images, statically generated websites or single page applications (SPA).

Elastic Load Balancer

Deliver applications hosted behind ELB such as dynamic websites, web services, and APIs.

API Gateway

Deliver API endpoints for REST APIs hosted on API Gateway.

Elemental MediaPackage

Deliver end-to-end live events or video on demand (VOD).

VPC origin

Deliver applications and content hosted within private VPCs, such as EC2 instances and Application Load Balancers.

Other

Refer to any AWS or non-AWS origin through its publicly resolvable URL.

Origin

S3 origin

Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

[Browse S3](#)

Origin path - optional

The directory path within your origin where your content is stored. [Learn more](#)

Settings

CloudFront provides default origin and cache settings based on what origin you selected. [View default settings for S3](#)

Origin settings

Origin settings control how CloudFront connects to the specified origin.

Use recommended origin settings

Customize origin settings

Review and create

General configuration		Edit	
Distribution name	lizhu-static-site-demo-cf	Description	-
Origin			Edit
S3 origin	lizhu-static-site-demo-s3.s3-website-us-east-1.amazonaws.com	Origin path	-
Connection timeout	10	Enable Origin Shield	No
		Connection attempts	3
Cache settings			Edit
CloudFront will apply default cache settings tailored to serving content from a S3 origin. You can customize settings after you create your distribution.			
Security			Edit
Security protections	None	Use monitor mode	No
		Use existing WAF configuration	No
Cancel		Previous	Create distribution

lizhu-static-site-demo-cf

Standard

View metrics

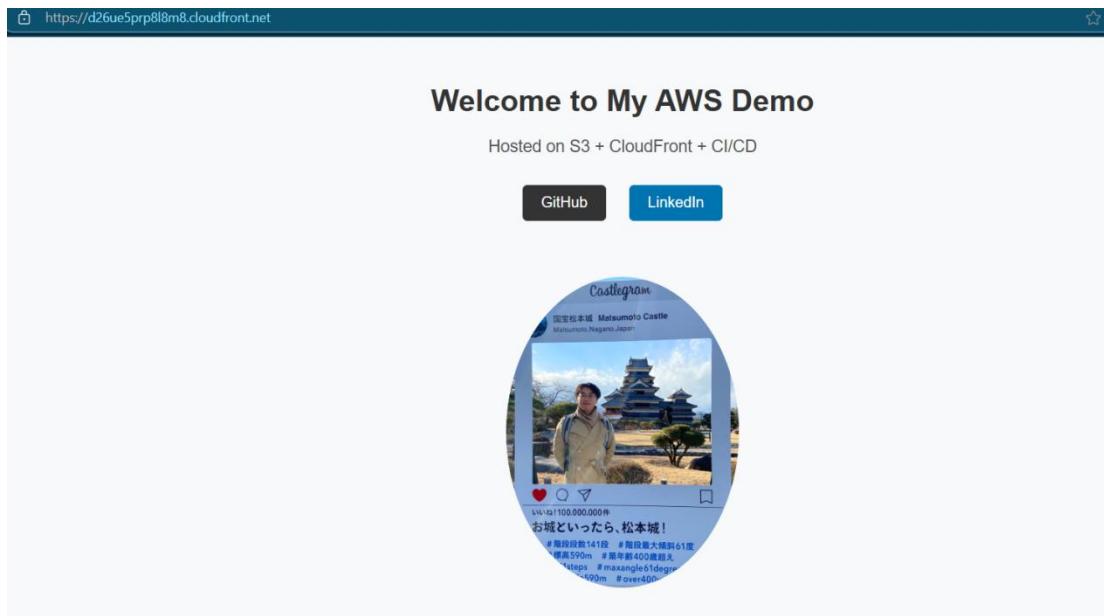
General

Cancer

Previous

Create distribution

Test site access via the CloudFront distribution domain.



Add three CloudFront Functions:

- Path rewriting
- Simple access control
- Caching optimization

[CloudFront](#) > [Functions](#) > AccessRewriteCacheFunction

AccessRewriteCacheFunction

[Edit](#)
[Delete](#)

Details	
Name	AccessRewriteCacheFunction
Development Runtime	cloudfront-js-2.0
ARN	arn:aws:cloudfront:█████████████████████:function:AccessRewriteCacheFunction
Description	This CloudFront Function implements three key features for a static website: 1. Access Control — Restricts access to specific paths to enhance security. 2. Path Rewriting — Automatically redirects root requests ("") to the default index page. 3. Cache Optimization — Adds a Cache-Control header to improve performance by enabling browser caching for a defined duration.
Live Runtime	-
Last modified	September 27, 2025 at 4:32:16 PM UTC

Associated KeyValueStore [Info](#)

No KeyValueStore associated with this function

[Associate existing KeyValueStore](#) [Create new KeyValueStore](#)

[CloudFront](#) > [Functions](#) > AccessRewriteCacheFunction

Function code

[Development](#)
[Live](#)
[Save changes](#)

```

1▼ function handler(event) {
2    var request = event.request;
3    var uri = request.uri;
4
5    // Access Control: Only allow specific paths
6    if (uri !== "/" && uri !== "/index.html" && uri !== "/profile.jpg") {
7        return {
8            statusCode: 403,
9            statusDescription: "Forbidden"
10        };
11    }
12
13    // Path Rewrite: Redirect "/" to "/index.html"
14    if (uri === "") {
15        request.uri = "/index.html";
16    }
17
18    // Cache Optimization: Set Cache-Control header to cache content for 1 hour
19    request.headers["cache-control"] = [
20        {
21            value: "max-age=3600"
22        }
23    ];
24
25    return request;
26}

```

JavaScript [Ln 25, Col 1](#) [Errors: 0](#) [Warnings: 0](#)

[CloudFront](#) > [Functions](#) > AccessRewriteCacheFunction

[Build](#)
[Test](#)
[Publish](#)
[Unpublished](#)

Test function

Use the form to create a test event. You can test the function without saving the test event, or you can save up to 10 test events per function.

Select test event

[+ New test event](#)

Event type The event type to test.	Stage Function stage to test.
Viewer request	Development

Request

HTTP method	URL path
GET	/li.html
IP address IP address of the request.	203.0.113.1
Request headers - optional	
Header	Value
host	example.cloudfront.net

[Add header](#) [Edit JSON](#) [Save](#) [Test function](#)

Edit JSON

```
1 [{}  
2   "version": "1.0",  
3   "context": {  
4     "eventType": "viewer-request"  
5   },  
6   "viewer": {  
7     "ip": "203.0.113.1"  
8   },  
9   "request": {  
10    "method": "GET",  
11    "uri": "/li.html",  
12    "headers": {  
13      "host": {  
14        "value": "example.cloudfront.net"  
15      }  
16    },  
17    "cookies": {},  
18    "queryString": {}  
19  }  
20 ]
```

JSON Ln 1, Col 1 Errors: 0 | Warnings: 0

Cancel Save

Execution result

Status 403 Forbidden Stage DEVELOPMENT Compute utilization Info 4

Output

```
{  
  "response": {  
    "statusCode": 403,  
    "statusDescription": "Forbidden",  
    "headers": {},  
    "cookies": {}  
  }  
}
```

Execution logs

CloudFront > Functions > AccessRewriteCacheFunction

Test function

Use the form to create a test event. You can test the function without saving the test event, or you can save up to 10 test events per function.

Select test event

+ New test event

Event type
The event type to test.

Viewer request

Stage
Function stage to test.

Development

Request

HTTP method GET

URL path /

IP address
IP address of the request.

203.0.113.1

Request headers - optional

Header	Value
host	example.cloudfront.net

Add header

Request cookies - optional

Execution result

Status Succeeded

Stage DEVELOPMENT

Compute utilization Info 4

Output

```
{
  "request": {
    "method": "GET",
    "uri": "/index.html",
    "querystring": {},
    "headers": {
      "cache-control": {
        "value": "max-age=3600"
      },
      "host": {
        "value": "example.cloudfront.net"
      }
    }
}
```

Execution logs

Build | Test | **Publish** Unpublished

Publish

Publish this function to copy it from the development stage to the live stage. Then you can associate the live function with one or more cache behaviors in your CloudFront distributions.

Publish function

Associate

WARNING: You are about to associate a function that will overwrite the existing function for the following cache behaviors. Are you sure you want to continue?

Distribution
E34NAZ639SZODO

Event type
Viewer request

Cache behavior
Choose options

Default (*) X

Publish function

View distribution **Remove association** **Add association**

< 1 > | **Associations** |

Cancel **Add association**

Add association

Build | Test | **Publish**

Publish

Publish this function to copy it from the development stage to the live stage. This also updates the following 1 cache behaviors to use the newly published function.

Publish function

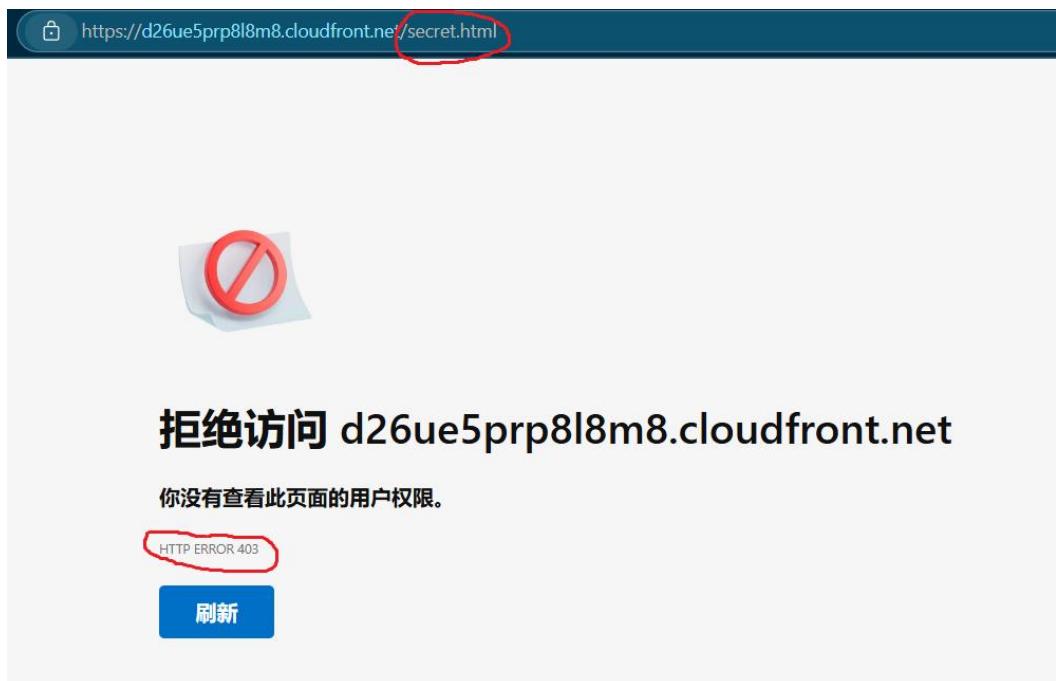
Associated distributions

Search distributions and cache behaviors

Distribution ID	Description	Cache behavior	Event type
E54NAZ6395ZOD0	-	Default (*)	Viewer Request

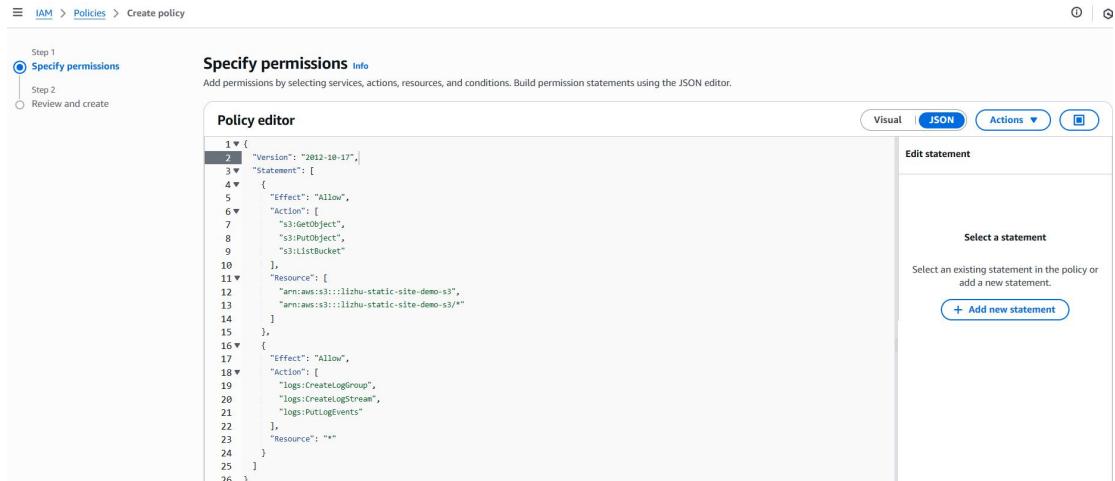
View distribution **Remove association** **Add association**

Test the function on the website



5. Create IAM Policy and CodeBuild Role

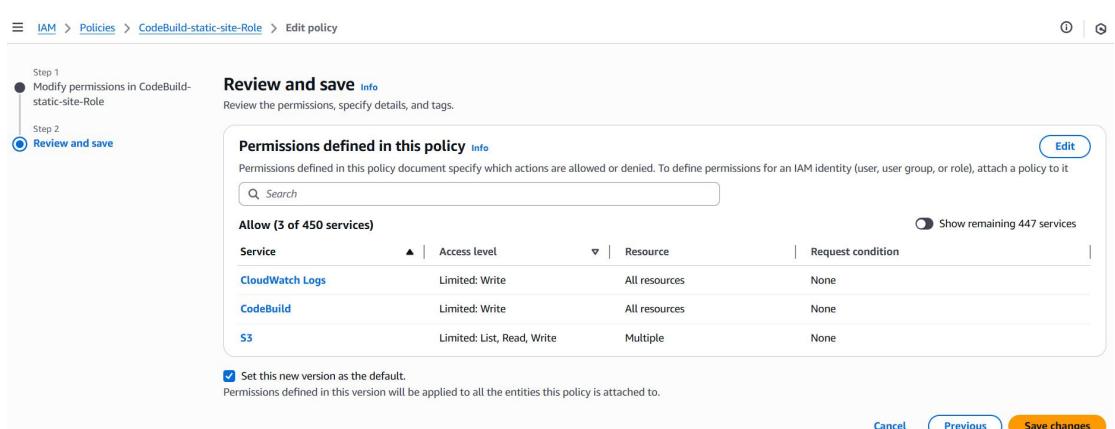
Define a custom IAM policy with least privilege permissions (read/write artifact bucket, full sync with Deploy Bucket, logging, reporting).



The screenshot shows the 'Specify permissions' step of creating a new IAM policy. The policy editor displays the following JSON code:

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "s3:GetObject",  
8                 "s3:PutObject",  
9                 "s3>ListBucket"  
10            ],  
11            "Resource": [  
12                "arn:aws:s3:::lizhu-static-site-demo-s3*",  
13                "arn:aws:s3:::lizhu-static-site-demo-s3/*"  
14            ]  
15        },  
16        {  
17            "Effect": "Allow",  
18            "Action": [  
19                "logs:CreateLogGroup",  
20                "logs:CreateLogStream",  
21                "logs:PutLogEvents"  
22            ],  
23            "Resource": "*"  
24        }  
25    ]  
26}
```

The 'Visual' tab is selected at the top right. A sidebar on the right shows a 'Select a statement' section with a '+ Add new statement' button.

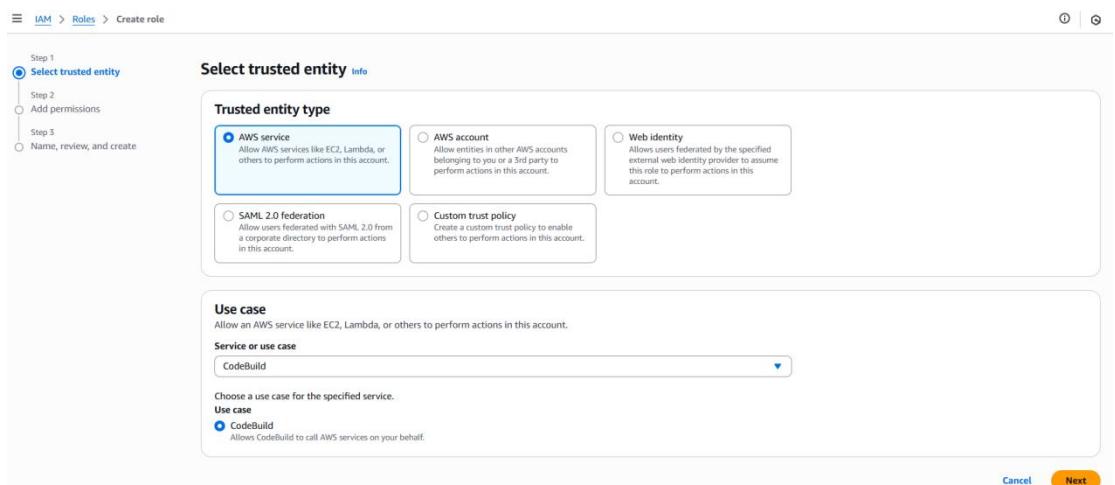


The screenshot shows the 'Review and save' step of creating a new IAM policy. It lists the permissions defined in the policy:

Service	Access level	Resource	Request condition
CloudWatch Logs	Limited: Write	All resources	None
CodeBuild	Limited: Write	All resources	None
S3	Limited: List, Read, Write	Multiple	None

A checkbox 'Set this new version as the default.' is checked. At the bottom are 'Cancel', 'Previous', and 'Save changes' buttons.

Attach the policy to a new IAM Role (CodeBuild-static-site) for use in the build process.



The screenshot shows the 'Select trusted entity' step of creating a new IAM role. It allows selecting the type of trusted entity:

- AWS service: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

The 'Use case' section indicates it's for CodeBuild. At the bottom are 'Cancel' and 'Next' buttons.

The screenshot shows the 'Add permissions' step of creating a new IAM role. On the left, a sidebar lists three steps: Step 1 (Select trusted entity), Step 2 (Add permissions, which is selected), and Step 3 (Name, review, and create). The main area is titled 'Add permissions' and shows a table of 'Permissions policies (1/1077)'. A search bar at the top right says 'Search' and has a dropdown set to 'Customer managed'. The table has columns for 'Policy name', 'Type', and 'Description'. One policy is selected: 'CodeBuild-static-site-Role'. The description for this policy is 'Minimal permissions for CodeBuild to do...'. At the bottom, there's a note '▶ Set permissions boundary - optional' and a 'Next' button.

The screenshot shows the 'Name, review, and create' step of creating a new IAM role. The sidebar shows Step 1, Step 2, and Step 3. The main area is titled 'Name, review, and create' and contains a 'Role details' section. It includes fields for 'Role name' (set to 'CodeBuild-static-site') and 'Description' (set to 'Minimal permissions for CodeBuild to deploy static website to S3 and log to CloudWatch'). Both fields have character limits and character count indicators.

6. Create the CodeBuild Project

Create a CodeBuild project with inline build commands.

The screenshot shows the 'Create build project' step in the AWS CodeBuild console. The sidebar shows 'Developer Tools > CodeBuild > Build projects > Create build project'. The main area is titled 'Create build project' and has a 'Project configuration' section. It includes a 'Project name' field (set to 'lizhu-static-site-demo'), a 'Project type' section with two options ('Default project' selected, 'Runner project' available), and an 'Additional configuration' section with a note about description, public build access, build badge, concurrent build limit, and tags.

Source 1 - Primary

Source provider

GitHub



Credential

Your account is successfully connected through OAuth using CodeBuild managed token. [Manage account credentials.](#)

Use override credentials for this project only

Repository

Repository in my GitHub account

Public repository

GitHub scoped webhook

Repository URL

<https://github.com/lizh1994/aws-static-site-cicd-lizhu>

<https://github.com/<user-name>/<repository-name>>

Source version - *optional info*

Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

► Additional configuration

Git clone depth, Git submodules

aws | Search [Alt+S]

☰ Manage default source credential

Source Provider

Credential type

GitHub App
Connect project to GitHub using an AWS managed GitHub App

Personal access token
Connect project to GitHub using a personal access token

OAuth app
Connect project to GitHub using an OAuth app

Service

Secrets Manager (recommended)
Use Secrets Manager to store token

CodeBuild
Use CodeBuild managed token

Connect to GitHub

▼ Environment

Provisioning model Info

On-demand

Automatically provision build infrastructure in response to new builds.

Reserved capacity

Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image

Managed image

Use an image managed by AWS CodeBuild

Custom image

Specify a Docker image

Compute

EC2

Optimized for flexibility during action runs

Lambda

Optimized for speed and minimizes the start up time of workflow actions

Running mode

Container

Running on Docker container

Instance

Running on EC2 instance directly



Search  [Alt+S]

Runtime(s)

Standard 

Image

aws/codebuild/amazonlinux-x86_64-standard:5.0 

Image version

Always use the latest image for this runtime version 

Use GPU-enhanced compute

Service role

New service role

Create a service role in your account

Existing service role

Choose an existing service role from your account

Role ARN

 arn:aws:iam:XXXXXXXXXX:role/CodeBuild-static-site 

Allow AWS CodeBuild to modify this service role so it can be used with this build project.

► Additional configuration

Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

▼ **Buildspec**

Build specifications

Insert build commands
Store build commands as build project configuration

Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Build commands [Info](#)  

```
1 version: 0.2
2 phases:
3   install:
4     runtime-versions:
5       python: 3.9
6   build:
7     commands:
8       - echo "Build started on `date`"
9       - aws s3 sync src/ s3://lizhu-static-site-demo-s3/ --delete
10
```

Initially, no artifact output was configured.

▼ **Artifacts** [Add artifact](#)

Artifact 1 - Primary

Type

No artifacts 

You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

► **Additional configuration**
Cache, encryption key

▼ **Logs**

CloudWatch

CloudWatch logs - optional
Checking this option will upload build output logs to CloudWatch.

Group name - optional

aws/codebuild/lizhu-static-site-demo

The group name of the logs in CloudWatch Logs. The log group name will be /aws/codebuild/<project-name> by default.

Test the build: confirmed that updated files were written to the Deploy Bucket.

lizhu-static-site-demo-s3 [Info](#)

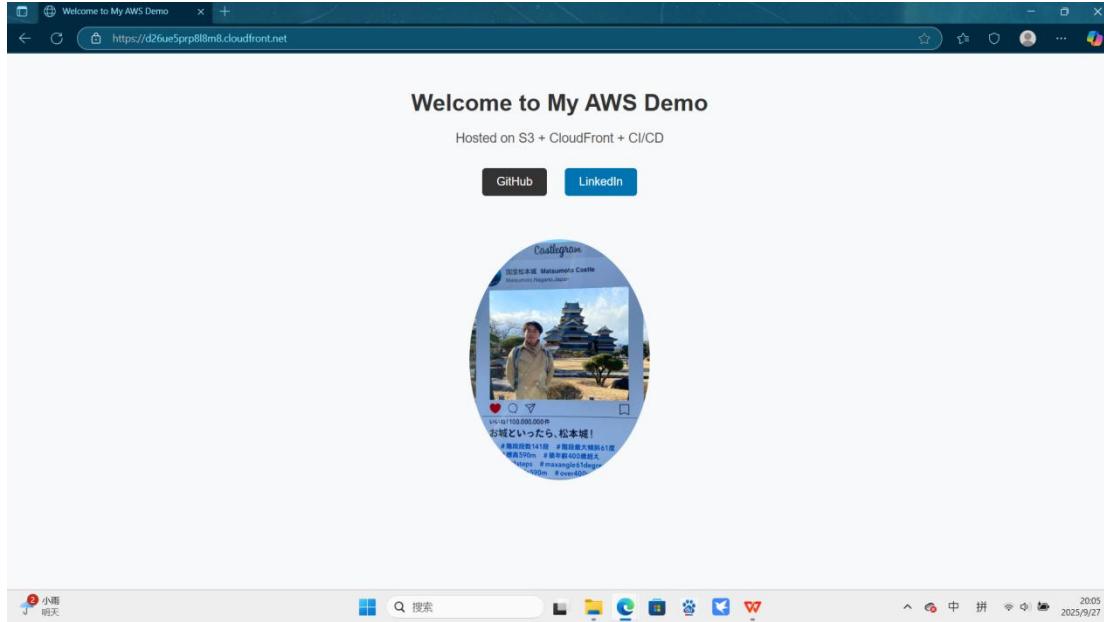
[Objects](#) | [Metadata](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2)

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](#)

Find objects by prefix (C) Copy S3 URI (C) Copy URL Download Open Delete Actions Create folder Upload

<input type="checkbox"/> Name	Type	Last modified	Size	Storage class
<input type="checkbox"/> index.html	html	September 27, 2025, 20:03:16 (UTC+01:00)	1.5 KB	Standard
<input type="checkbox"/> profile.jpg	jpg	September 27, 2025, 20:03:16 (UTC+01:00)	3.5 MB	Standard



7. Create CodePipeline

Update the CodeBuild project with build commands that generate build artifacts for the Deploy stage.

▼ **Buildspec**

Build specifications

Insert build commands
Store build commands as build project configuration

Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Build commands [Info](#)  

```
1 version: 0.2
2 phases:
3   install:
4     runtime-versions:
5       python: 3.9
6   build:
7     commands:
8       - echo "Build started on `date`"
9       - aws s3 sync src/ s3://lizhu-static-site-demo-s3/ --delete
10  artifacts:
11    files:
12      - '**/*'
```

✖ 0 ⚠ 0 12:13 YAML

Create the CodePipeline with three stages:

- Source (GitHub) - Trigger: push events.
- Build (CodeBuild) - Uses the updated CodeBuild project.
- Deploy (S3) - Deploys artifacts to the Deploy Bucket.

☰ [Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > Create new pipeline

Step 1
Choose creation option [Info](#)
Step 1 of 7

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

Category

Deployment Continuous Integration Automation
 Build custom pipeline

[Cancel](#) [Next](#)

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose pipeline settings Info

Step 2 of 7

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Execution mode Info
Choose the execution mode for your pipeline. This determines how the pipeline is run.
 Superseded
 Queued
 Parallel

Service role
 New service role
Create a service role in your account
 Existing service role
Choose an existing service role from your account

Role name

Type your service role name
 Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add source stage Info

Step 3 of 7

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Information
The GitHub (via OAuth app) action is not recommended
 The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (via GitHub App) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository

Branch

Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Next](#)

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose creation option

Step 2 Choose pipeline settings

Step 3 Add source stage

Step 4 Add build stage

Step 5 Add test stage

Step 6 Add deploy stage

Step 7 Review

Add source stage Info

Step 3 of 7

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:us-east-1 or [Connect to GitHub](#)

Repository name
Choose a repository in your GitHub account.

lizh1994/aws-static-site-cicd-lizhu

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

main

Output artifact format
Choose the output artifact format.

CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions. [Learn more](#)

Enable automatic retry on stage failure

Webhook events

Webhook - optional

Start your pipeline on push and pull request events.

► Webhook event filters - optional

Starts your pipeline on a specific event.

[Remove filters](#)

Step 1
[Choose creation option](#)

Step 2
[Choose pipeline settings](#)

Step 3
[Add source stage](#)

Step 4
Add build stage

Step 5
[Add test stage](#)

Step 6
[Add deploy stage](#)

Step 7
[Review](#)

Add build stage [Info](#)

Step 4 of 7

Build - optional

Build provider
 Choose the tool you want to use to run build commands and specify artifacts for your build action.

Commands Other build providers

AWS CodeBuild ▾

Project name
 Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

lizhu-static-site-demo or

Define buildspec override - *optional*
 Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

Environment variables - optional
 Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Build type

Single build
 Triggers a single build. Batch build
 Triggers multiple builds as a single execution.

Region
 ▾

Input artifacts
 Choose an input artifact for this action. [Learn more](#)

Defined by: Source

Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Skip build stage](#) **Next**

Step 1 [Choose creation option](#)

Step 2 [Choose pipeline settings](#)

Step 3 [Add source stage](#)

Step 4 [Add build stage](#)

Step 5 [Add test stage](#)

Step 6 [Add deploy stage](#)

Step 7 [Review](#)

Add deploy stage Info

Step 6 of 7

Deploy - optional

Deploy provider
Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for that provider.

Amazon S3 ▼

Region
United States (N. Virginia) ▼

Input artifacts
Choose an input artifact for this action. [Learn more](#) ▼

BuildArtifact X
Defined by: Build

No more than 100 characters

Bucket
lizhu-static-site-demo-s3 X

Deployment path - optional
/

Extract file before deploy
The deployed artifact will be unzipped before deployment.

Additional configuration

Configure automatic rollback on stage failure

Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Skip deploy stage](#) **Next**

Step 1 [Choose creation option](#)

Step 2 [Choose pipeline settings](#)

Step 3 [Add source stage](#)

Step 4 [Add build stage](#)

Step 5 [Add test stage](#)

Step 6 [Add deploy stage](#)

Step 7 [Review](#)

Review Info

Step 7 of 7

Step 2: Choose pipeline settings

Pipeline settings

Pipeline name
lizhu-static-site-demo

Pipeline type
V2

Execution mode
SUPERSEDED

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo

Step 3: Add source stage

Source action provider

Source action provider
GitHub (via GitHub App)

OutputArtifactFormat
CODE_ZIP

DetectChanges
true

ConnectionArn
arn:aws:codeconnections:us-east-1:370970796707:connection/0340bf7e-ef52-49a9-bbaa-b2c4a54e7685

FullRepositoryId
lizhu1994/aws-static-site-cicd-lizhu

Default branch
main

Enable automatic retry on stage failure
Disabled

Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type

No filter

Step 4: Add build stage

Build action provider

Build action provider
AWS CodeBuild

ProjectName
lizhu-static-site-demo

Commands
-

Enable automatic retry on stage failure
Enabled

Step 6: Add deploy stage

Deploy action provider

Deploy action provider
Amazon S3

Extract
true

BucketName
lizhu-static-site-demo-s3

ObjectKey
/

Configure automatic rollback on stage failure
Enabled

Enable automatic retry on stage failure
Disabled

Screenshot of the AWS CodePipeline console showing the pipeline "lizhu-static-site-demo". The pipeline consists of three stages: Source (GitHub), Build (AWS CodeBuild), and Deploy (Amazon S3). The Source stage is successful, while the Build and Deploy stages are in progress.

The screenshot also shows the AWS Lambda console with a search bar for "Search" and the Amazon S3 console showing buckets "codepipeline-us-east-1-e1503f64bcd9-42cd-8777-45da75ae8e75" and "lizhu-static-site-demo-s3".

Note: Pipeline Artifact Bucket was automatically created by AWS to handle intermediate artifacts.

8. Troubleshooting

The first pipeline execution will always fail because the pipeline artifact bucket is just created, and the CodeBuild role does not yet have permissions for it.

Screenshot of the AWS CodePipeline console showing the execution summary for the pipeline "lizhu-static-site-demo". The execution failed due to a permission issue with the AWS CodeBuild role. The latest action message indicates that the user lacks the necessary permissions to start the build.

Step 1: Fix Initial Failure

- Go to the CodeBuild IAM Role.
- Add permissions for the newly created pipeline artifact bucket (Read/Write/List).
- Save and re-deploy the pipeline.

The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b. The bucket details page is displayed, showing a single object named 'lizhu-static-site-de/' which is a folder.

The screenshot shows the AWS IAM Policy editor for the 'CodeBuild static site Role'. The policy document is being edited, showing two statements. The first statement grants 'Allow' access to S3 actions (GetObject, PutObject, ListBucket) on resources starting with 'arn:aws:s3:::lizhu-static-site-demo-s3/*'. The second statement grants 'Allow' access to CloudWatch Logs actions (CreateLogGroup, CreateLogStream, PutLogEvents) on resources starting with 'arn:aws:s3:::codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b/*'. The right sidebar shows available services like API Gateway and CloudWatch Metrics.

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": [
7          "s3:GetObject",
8          "s3:PutObject",
9          "s3>ListBucket"
10         ],
11        "Resource": [
12          "arn:aws:s3:::lizhu-static-site-demo-s3",
13          "arn:aws:s3:::lizhu-static-site-demo-s3/*",
14          "arn:aws:s3:::codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b",
15          "arn:aws:s3:::codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b/*"
16        ]
17      },
18      {
19        "Effect": "Allow",
20        "Action": [
21          "logs>CreateLogGroup",
22          "logs>CreateLogStream",
23          "logs>PutLogEvents"
24        ],
25      }
26    ],
27  }
  
```

The screenshot shows the 'Review and save' step in the IAM Policy editor. It displays the permissions defined in the policy and allows setting a new version as the default. At the bottom, there are 'cancel', 'Previous', and 'Save changes' buttons.

Permissions defined in this policy

Service	Access level	Resource	Request condition
CloudWatch Logs	Limited: Write	All resources	None
CodeBuild	Limited: Write	All resources	None
S3	Limited: List, Read, Write	Multiple	None

Set this new version as the default.

Permissions defined in this version will be applied to all the entities this policy is attached to.

cancel Previous Save changes

The screenshot shows the IAM Roles page with the 'CodeBuild static site Role' selected. A green banner at the top indicates it was updated. The role details show it is customer-managed and has an ARN. The 'Permissions' tab is selected, showing the attached policy.

Policy details

Type	Customer managed	Creation time	September 27, 2025, 18:42 (UTC+01:00)
		Edited time	September 28, 2025, 11:08 (UTC+01:00)
		ARN	arn:aws:iam::[REDACTED]:policy/Co
			deBuild-static-site-Role

Permissions

Step 2: If Issues Persist

- CodeBuild permissions: Ensure the role includes full access (Get/Put/Delete/List) to the deploy S3 bucket and necessary access to the pipeline bucket.
- CodePipeline permissions: If the pipeline cannot pass artifacts between stages, add an inline policy to the CodePipeline role with the required s3:* actions on the artifact bucket.

The screenshot shows the AWS CodePipeline execution details for a pipeline named "lizhu-static-site-demo". The execution ID is "8d591133". The pipeline consists of three stages: Source, Build, and Deploy. The Source stage (GitHub via OAuth app) and Build stage (AWS CodeBuild) both show "All actions succeeded". The Deploy stage shows "1 of 1 action failed" due to a "Deploy Amazon S3" step failing. The execution summary indicates a status of Failed, started 5 minutes ago, completed 5 minutes ago, and a duration of 39 seconds. A red box highlights the latest action execution message, which states: "You are missing permissions to call s3.getObject on the input artifact, codepipeline-us-east-1-e1503f64bcd9-42cd-8777-45da75ae8e75/lizhu-static-site-de/BuildArtif/xvcuAzK. Verify that the policy or the resource allows you to perform this task: User: arnawssts:assumed-role/AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo/1759007210255 is not authorized to perform: s3:listBucket on resource: 'arnaws:s3:::codepipeline-us-east-1-e1503f64bcd9-42cd-8777-45da75ae8e75'" because no identity-based policy allows the s3:listBucket action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: YPYD5CPH7Z582YA8; S3 Extended Request ID: n4u9BhFo7XBa0kCvrlD1kQMkBWc29+px/MExnKyb1YhdniE+GrwQ8hSqlmYL35y5Cq+PCE+mONnNDRDqHCU10kU9onaayV+rmlVCzrEWw; Proxy: null)".

The screenshot shows the IAM Roles page for the "AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo". The role was created on September 27, 2025, at 20:34 UTC. It has a maximum session duration of 1 hour. The Permissions tab shows three attached policies: "AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo" (Customer managed), "CodePipeline-CodeBuild-us-east-1-lizhu-st..." (Customer managed), and "CodePipeline-CodeBuild-us-east-1-lizhu-st..." (Customer managed). The ARN is listed as "arn:aws:iam::[REDACTED]:role/service-role/AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo".

IAM > Roles > AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo

Identity and Access Management (IAM)

AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo

Summary

Creation date: September 27, 2025, 20:34 (UTC+01:00)

ARN: arn:aws:iam::██████████:role/service-role/AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo

Last activity: 11 minutes ago

Maximum session duration: 1 hour

Permissions **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

Permissions policies (4) Info

You can attach up to 10 managed policies.

Add permissions **Simulate** **Remove**

Attached entities

Policy name	Type	Attached entities
AWSCodePipelineServiceRole-us-east-1-liz...	Customer managed	1
CodePipeline-CodeBuild-us-east-1-lizhu-st...	Customer managed	1

IAM > Roles > AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo > Create policy

Specify permissions **Info**

Step 1: Specify permissions Step 2: Review and create

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

1 "Version": "2012-10-17",
2   "Statement": [
3     {
4       "Effect": "Allow",
5       "Action": [
6         "s3:GetObject",
7         "s3:PutObject",
8         "s3:ListBucket"
9       ],
10      "Resource": [
11        "arn:aws:s3:::codepipeline-us-east-1-e1503f64bcd9-42cd-8777-45da75ae8e75",
12        "arn:aws:s3:::codepipeline-us-east-1-e1503f64bcd9-42cd-8777-45da75ae8e75/*"
13      ]
14    }
15  ]
16 ]
17

```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

Developer Tools > CodePipeline > Pipelines > lizhu-static-site-demo > 54b706ff

Pipeline execution: 54b706ff

Source All actions succeeded. GitHub (via GitHub API) 4 minutes ago

Build All actions succeeded. AWS CodeBuild 4 minutes ago

Deploy Amazon S3 4 minutes ago

Rerun **Stop execution** **< Previous execution** **Next execution >**

Summary **Timeline** **Variables** **Revisions** **Stage**

Execution summary

Status	Started	Completed	Duration
Failed	4 minutes ago	4 minutes ago	41 seconds

Trigger StartPipelineExecution - AdminUI

Pipeline execution ID 54b706ff-94d0-4aa3-9b93-13166b1c6609

Latest action execution message

You do not have sufficient permissions to call s3:putObject for the deployment bucket, lizhu-static-site-demo->s3. Verify that the policy on the resource allows you to perform this task. If you choose a canned ACL for your Amazon S3 deployment action, the policy must include the PutObjectAcl action. If the object already exists, the policy must also include the PutObjectVersionAcl action. User: arn:aws:sts::██████████:assumed-role/AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo/1759054670694 is not authorized to perform: s3:PutObject because no identity-based policy allows the s3:PutObject action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 08073298WM2CR0F; S3 Extended Request ID: k29GnTiyJalGao5n6OGO+d992xpqL89/YXXmnG9wLAjPMH/SUqzBQdgIffFWVmPqlQBoTWQEgeSNjRAgMTa/10hDgkxZl419lymmMgPwQ; Proxy: null)

Diagnose with Amazon Q

IAM > Roles > AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo > Create policy

Step 1 Specify permissions Step 2 Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": [
7                  "s3:PutObject",
8                  "s3:PutObjectAcl",
9                  "s3:PutObjectVersionAcl"
10             ],
11             "Resource": "arn:aws:s3:::lizhu-static-site-demo-*"
12         },
13         {
14             "Effect": "Allow",
15             "Action": "s3>ListBucket",
16             "Resource": "arn:aws:s3:::lizhu-static-site-demo-*"
17         }
18     ]
19 }
```

IAM > Roles > AWSCodePipelineServiceRole-us-east-1-lizhu-static-site-demo > Create policy

Step 1 Specify permissions Step 2 Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
`CodePipelineServiceRole-S3-extra`
Maximum 128 characters. Use alphanumerics and '-' characters.

Permissions defined in this policy Info
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Service	Access level	Resource	Request condition
S3	Limited: List, Permissions management, Write	Multiple	None

Show remaining 449 services

Search

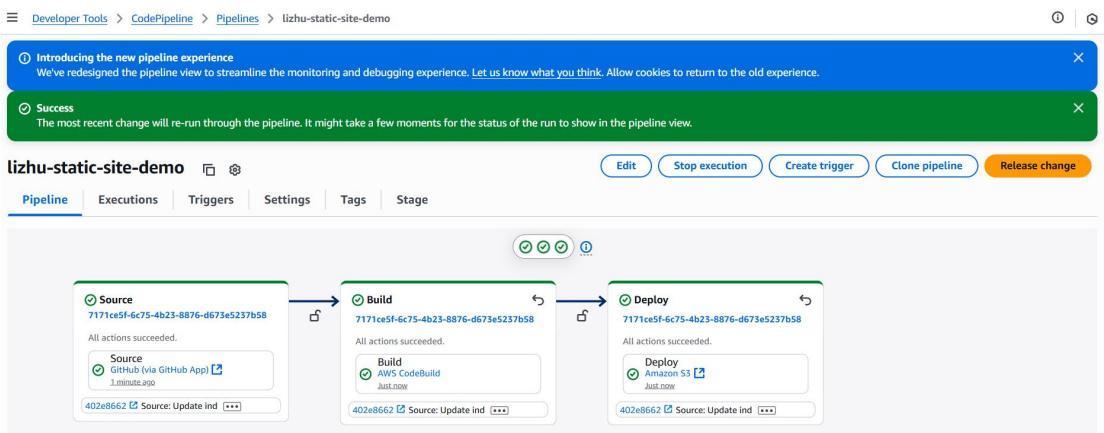
Edit

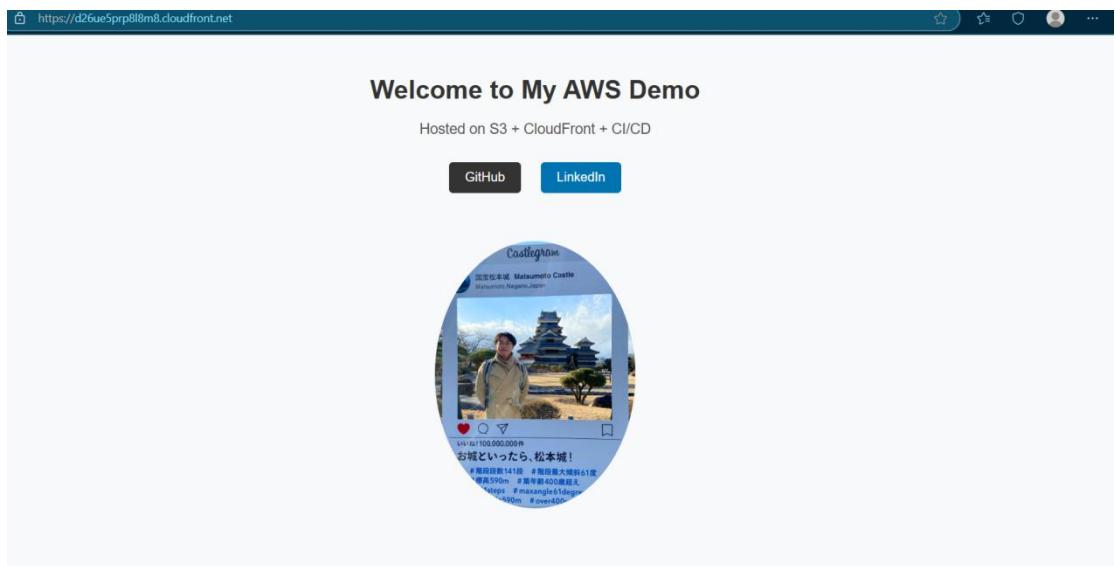
Cancel Previous **Create policy**

Most failures at this stage are caused by IAM misconfigurations or artifact exchange issues between services.

9. Validate Deployment

After successful deployment, test the CloudFront distribution to confirm site updates.





Modify `index.html`, commit the change to GitHub, and confirm the pipeline automatically redeploys the updated version. I replaced LinkedIn by Wikipedia, and you can find in the following screenshots the website changed.

Developer Tools > CodePipeline > Pipelines > lizhu-static-site-demo

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.

lizhu-static-site-demo ⋮

Pipeline Executions Triggers Settings Tags Stage

Source 2c4820f5-8f8f-4aa9-b7b3-139b7ae7895a
All actions succeeded.
Source GitHub (via GitHub App) Just now
f2ab9eaa Source: Update ind [...]

Build 2c4820f5-8f8f-4aa9-b7b3-139b7ae7895a
All actions succeeded.
Build AWS CodeBuild Just now
f2ab9eaa Source: Update ind [...]

Deploy 2c4820f5-8f8f-4aa9-b7b3-139b7ae7895a
All actions succeeded.
Deploy Amazon S3 Just now
f2ab9eaa Source: Update ind [...]

CloudFront <

Distributions
Policies
Functions
Static IPs
VPC origins
What's new

SaaS
Multi-tenant distributions

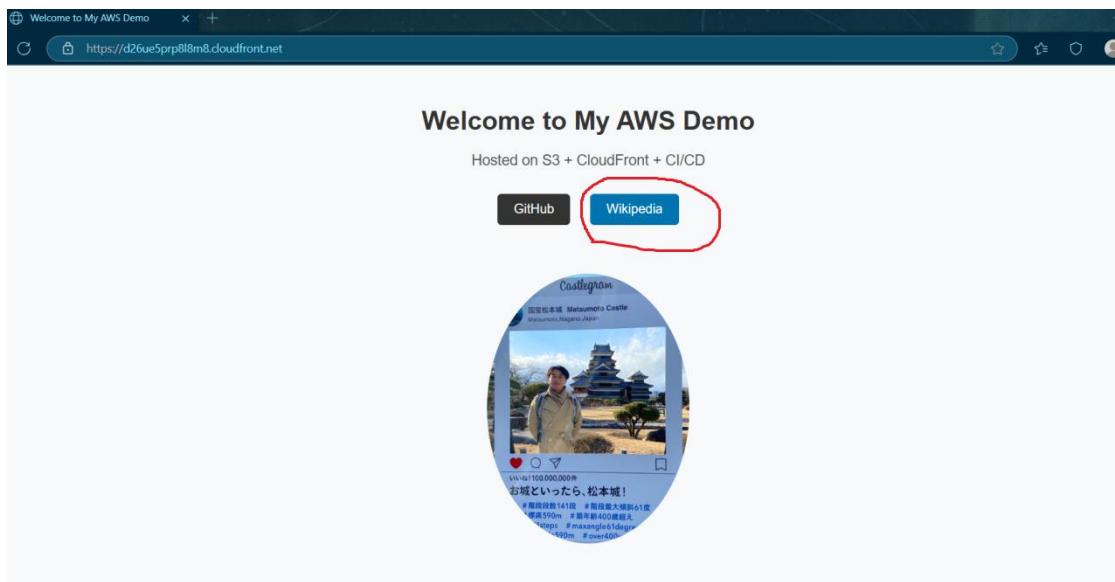
Successfully created invalidation I4M42CR0POTBK9YODFCFOXPN3.

Invalidation details

Date created
September 28, 2025 at 10:44:20 AM UTC

Status
Completed

Object paths
/*



10. Add Monitoring & Notifications

Create an SNS Topic and subscribe a personal email.

The screenshot shows the 'Create topic' page for Amazon Simple Notification Service (SNS). At the top, there's a blue header bar with a 'New Feature' message about High Throughput FIFO topics. Below the header, the page title is 'Amazon Simple Notification Service' with a subtitle 'Pub/sub messaging for microservices and serverless applications.' A description of SNS as a highly available, durable, secure, fully managed pub/sub messaging service follows. The main content area is titled 'Create topic' and contains a 'Topic name' input field with the value 'pipeline-failure-topic'. There are two options for topic type: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected. Both options have associated bullet points. Below the topic name input is a 'Name' input field containing 'pipeline-failure-topic'. A note says 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).' Below that is a 'Display name - optional' input field with the value 'My Topic'. A note says 'To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.' At the bottom, there are 'Next step' and 'Start with an overview' buttons.

New Feature
Amazon SNS now supports High Throughput FIFO topics. [Learn more](#)

Application Integration

Amazon Simple Notification Service

Pub/sub messaging for microservices and serverless applications.

Amazon SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and event-driven services.

≡ [Amazon SNS](#) > [Topics](#) > Create topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS

Standard

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Name

pipeline-failure-topic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional [Info](#)

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

My Topic

Maximum 100 characters.

New FeatureAmazon SNS now supports High Throughput FIFO topics. [Learn more](#)**Create subscription****Details****Topic ARN**

arn:aws:sns:us-east-1:████████████████████:pipeline-failure-topic

**Protocol**

The type of endpoint to subscribe

Email

**Endpoint**

An email address that can receive notifications from Amazon SNS.

test@example.com

Info After your subscription is created, you must confirm it. [Info](#)**AWS Notification - Subscription Confirmation**

AWS Notifications<no-reply@sns.amazonaws.com>



周日 2025/9/28 22:28

收件人: 你

此消息的语言为 英语

翻译至 始终不翻译 英语

You have chosen to subscribe to the topic:

arn:aws:sns:us-east-1:████████████████████:pipeline-failure-topic

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

答复

转发

New FeatureAmazon SNS now supports High Throughput FIFO topics. [Learn more](#)**Subscriptions (1)**[Edit](#)[Delete](#)[Request confirmation](#)[Confirm subscription](#)[Create subscription](#)[Search](#)

< 1 > ⌂

ID	Endpoint	Status	Protocol	Topic
45a4f280-f0f1-46a9-9c1f-f...	████████████████████	Confirmed	EMAIL	pipeline-failure-topic

Set up EventBridge to detect pipeline failure events.

Application Integration

Amazon EventBridge

A serverless service for building event-driven applications

Amazon EventBridge is a serverless service that uses events to connect application components together, making it easier for developers to build scalable event-driven applications.

Get started **EventBridge Rule**

A rule matches incoming events and sends them to targets for processing.

 EventBridge Pipes

A pipe connects an event source to a target with optional filtering and enrichment.

 EventBridge Schedule

A schedule invokes a target one-time or at regular intervals defined by a cron or rate expression.

 EventBridge Schema registry

Schema registries collect and organize schemas.

[Create rule](#)

Step 1 **Define rule detail**

- Step 2 Build event pattern
- Step 3 Select target(s)
- Step 4 - optional Configure tags
- Step 5 Review and create

Define rule detail Info

Rule detail

Name
pipeline-failure
Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-_.

Description - optional

Event bus Info
Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.

Enable the rule on the selected event bus

Rule type Info

- Rule with an event pattern
A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.
- Schedule
A rule that runs on a schedule

Cancel **Next**

Step 1 **Define rule detail**

- Step 2 **Build event pattern**
- Step 3 Select target(s)
- Step 4 - optional Configure tags
- Step 5 Review and create

Build event pattern Info

Events

You don't have to select or enter a sample event, but it's recommended so you can reference it when writing and testing the event pattern, or filter criteria.

Event source
Select the event source from which events are sent.

- AWS events or EventBridge partner events
Events sent from AWS services or EventBridge partners.
- Other
Custom events or events sent from more than one source, e.g. events from AWS services and partners.
- All events
All events sent to your account.

Sample event - optional
You don't have to select or enter a sample event, but it's recommended so you can reference it when writing and testing the event pattern, or filter criteria.

Event pattern Info

Creation method

- Use schema
Use an Amazon EventBridge schema to generate the event pattern.
- Use pattern form
Use a template provided by EventBridge to create an event pattern.
- Custom pattern (JSON editor)
Write an event pattern in JSON.

Event source
AWS service or EventBridge partner as source

AWS service
The name of the AWS service as the event source

Event type
The type of events as the source of the matching pattern

Event Type Specification 1

- Any state
- Specific state(s)

Specific state(s)

Event pattern
Event pattern, or filter to match the events

```
1 {
2   "source": ["aws.codepipeline"],
3   "detail-type": ["CodePipeline Pipeline Execution State Change"],
4   "detail": {
5     "state": ["FAILED"]
6   }
7 }
```

Copy **Test pattern** **Edit pattern**

Cancel **Previous** **Next**

Create rule

Step 3 **Select target(s)**

Step 4 - optional
Configure tags

Step 5
Review and create

Target 1

Target types
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.
 EventBridge event bus
 EventBridge API destination
 AWS service

Select a target | Info
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Target location
 Target in this account
 Target in another AWS account

Topic

Permissions
 Use execution role (recommended)

Execution role
EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so. [EventBridge and AWS Identity and Access Management](#)
 Create a new role for this specific resource
 Use existing role

Role name

Additional settings

Amazon EventBridge > Rules

Rule pipeline-failure was created successfully

Rules
A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

Select event bus

Event bus
Select or enter event bus name

Rules on default event bus (1)

Name	Status	Type	Event bus	ARN	Description
pipeline-failure	Enabled	Standard	default	arn:aws:events:us-east-1:123456789012:rule/pipeline-failure	-

Tested by intentionally breaking IAM permissions, I removed the s3:DeleteObject here → confirmed alert email was received.

Step 1 **Modify permissions in CodeBuild-static-site-Role**

Step 2 Review and save

Modify permissions in CodeBuild-static-site-Role Info

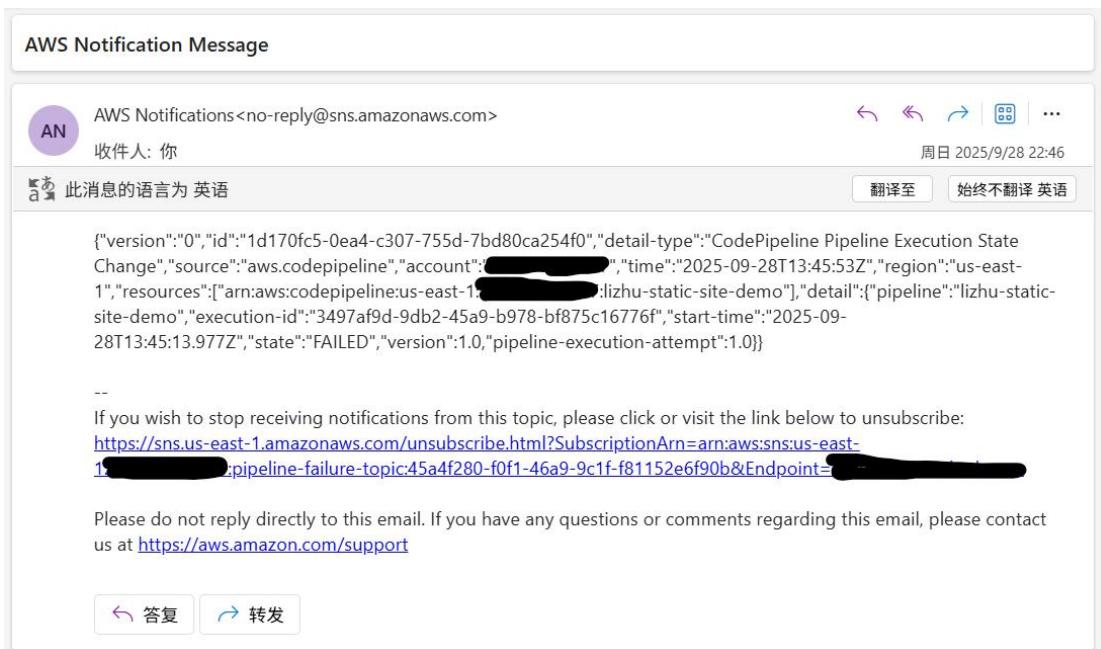
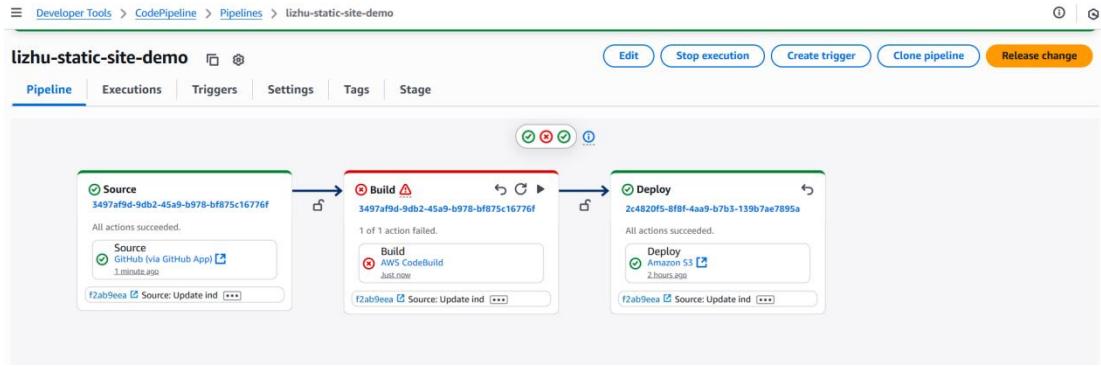
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

10    1,
11    "Resource": [
12      "arn:aws:s3:::codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b",
13      "arn:aws:s3:::codepipeline-us-east-1-cb9e224fb649-408a-b508-9043b93f978b/*"
14    },
15  },
16  {
17    "Effect": "Allow",
18    "Action": [
19      "s3:GetObject",
20      "s3:PutObject",
21      "s3:DeleteObject",
22      "s3>ListBucket"
23    ],
24    "Resource": [
25      "arn:aws:s3:::lizhu-static-site-demo-s3",
26      "arn:aws:s3:::lizhu-static-site-demo-s3/*"
27    ]
28  },
29  {
30    "Effect": "Allow",
31    "Action": [

```



11. Lessons Learned

11.1 IAM Permissions

During the project, I gained a much clearer understanding of how IAM roles and policies work. Striking the right balance between least privilege and task feasibility is essential:

- Roles must be granted just enough access to complete their tasks, but not more.
- This requires knowing exactly what each AWS service will do, and how it interacts with others.
- Permissions must also be scoped at the right level of granularity.

Many of my initial failures were caused by insufficient permissions. These issues forced me to gradually clarify the boundaries of each role's access, which deepened my understanding of IAM in practice.

11.2 Troubleshooting Skills

This was my first time building an AWS demo project end-to-end, so failures were

frequent. Through this process I learned how to:

- Read error logs carefully.
- Identify the root cause by correlating logs with service behavior.
- Apply targeted fixes instead of random trial and error.

Learning to debug AWS services systematically has been one of the most valuable outcomes of this project.

11. 3 Service Interactions

Another key lesson was to always map out how services exchange data and interact with each other. Missing or unclear interactions often caused failures.

For example, in my first pipeline runs the deploy stage failed repeatedly. After verifying permissions, I realized the real problem was that the build stage never produced a build artifact output. Without that artifact, CodePipeline could not hand anything over to the deploy stage.

Understanding these service interactions up front prevents wasted effort and unnecessary debugging.

Final Reflection

Most importantly, this project reminded me that hands-on practice is irreplaceable. Designing demos, facing failures, and solving problems along the way provided insights no tutorial could give me.

The key is to practice with curiosity, ask questions when things break, and keep the mindset that failure is part of the learning process.