

Static Website Deployment on AWS using Terraform

Course: DLBSEPCP01_E _Cloud Programming

Student Name: Kinyua Kennedy

Phase: Finalization phase- Portfolio part 3

Date of submission: 22/06/2025

Overview and Introduction

This project provisions a static website on Amazon S3 using Terraform for understanding fundamental cloud infrastructure concepts, complete with:

- Public access configuration
- Bucket policy
- Website configuration (index page)
- Uploading of HTML content

Project Structure

- main.tf # Terraform configuration for AWS resources
- variables.tf # Input variables like region and project name
- outputs.tf # Outputs like the website URL
- index.html # Your static website homepage
- README.md # This file

Features

- Creates an S3 bucket
- Enables website hosting
- Configures bucket policy for public read access
- Uploads a static index.html file
- Outputs the live website URL after deployment

Prerequisites

- Make sure you have the following installed:
- Terraform >= 1.0 and AWS CLI
- An AWS account with appropriate permissions
- AWS credentials configured (~/.aws/credentials or environment variables)

Tools used & Architecture Diagram

- **Terraform** – Infrastructure as code
- **AWS S3** – Stores static content
- **CloudFront** – Global CDN + HTTPS
- **IAM** – Access control and bucket policies

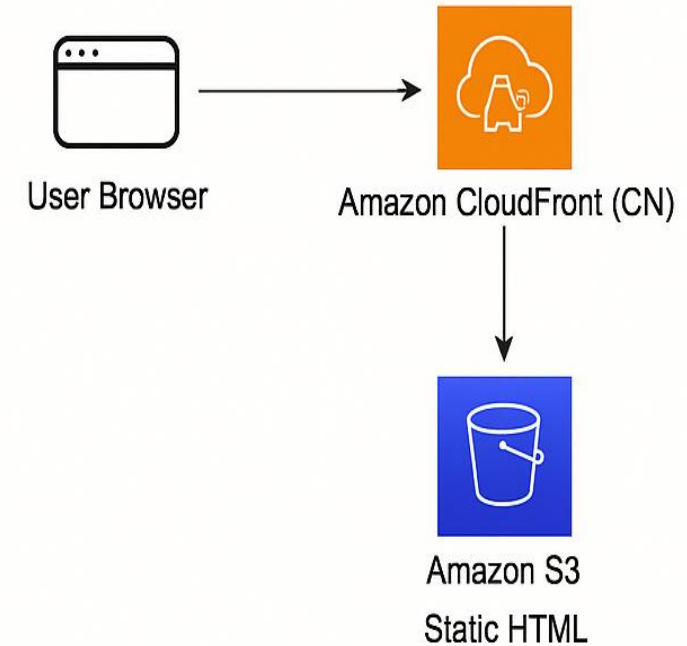
Terraform File Breakdown

- ***main.tf***: Defines S3, CloudFront, and IAM
- ***variables.tf***: Holds region/project settings
- ***outputs.tf***: Displays live URLs
- ***index.html***: My live webpage

This concept outlines the design for hosting a simple, highly available static website on AWS.

The static content will be served via Amazon S3 and distributed globally using Amazon CloudFront, providing low-latency access for users worldwide.

The entire infrastructure will be managed using Terraform, ensuring reproducible, version-controlled deployments.



Installation Manual: Setup Instructions

1. Clone the Repo

- git [clone https://github.com/your-username/static-website-terraform.git](https://github.com/your-username/static-website-terraform.git) cd static-website-terraform

2. Customize Variables

- Edit the variables.tf file or use Terraform.tfvars:
- project_name = "static-website" region = "eu-central-1"
- Alternatively, override on CLI:
- terraform apply -var="project_name=static-website" -var="region=eu-central-1"
- terraform apply
- Confirm with yes when prompted.

3. Initialize Terraform

- terraform init

4. Review the Plan

- terraform plan

5. Apply the Configuration

- terraform apply
- Confirm with yes when prompted.

Access my Website:

After a successful deployment, you'll see an output like that shown below

Click the URL or paste it into a browser to view the hosted website!

website_url = <http://static-website-bucket-kennedy-2025.s3-website.eu-central-1.amazonaws.com>

Key Concepts Used

- S3 Static Website Hosting
- Bucket Policy vs ACL (we use policy-based public access)
- Terraform Resource Dependencies
- Modular Configuration using Variables and Outputs

Security Considerations

- Public read access is enabled by policy, not ACLs, to comply with [Amazon S3 Object Ownership changes](#).
- Ensure you understand the implications of public access before deploying in production.

What Can Be Done Next

- Add custom domain support via Route 53
- Integrate CloudFront for HTTPS + CDN
- Add CI/CD via GitHub Actions
- Add versioning and logging

Challenges

- Initial attempt at Lambda + Docker failed (bootstrap errors)
- Pivoted to S3+CloudFront for simplicity and reliability
- Learned the value of adapting when roadblocks arise

Lessons Learnt

- Practical Terraform deployment
- Cloud resource configuration and IAM
- Simplicity often beats complexity
- Confidence using AWS for real-world scenarios

Terraform.tf state Script Documentation / Code Snippets

These files define the cloud infrastructure needed to host my static website.

Terraform is an Infrastructure as Code (IaC) tool that allows me to define and provision infrastructure using a declarative configuration language.

```
1  {
2    "version": 4,
3    "terraform_version": "1.12.1",
4    "serial": 16,
5    "lineage": "2d5ff0d4-a5d9-9d28-23d6-c1031333fd05",
6    "outputs": {
7      "cloudfront_url": {
8        "value": "https://dm7dyutyusru4.cloudfront.net",
9        "type": "string"
10     },
11     "website_url": {
12       "value": "http://static-website-bucket-kennedy-2025.s3-website.eu-central-1.amazonaws.com",
13       "type": "string"
14     }
15   },
16   "resources": [
17     {
18       "mode": "managed",
19       "type": "aws_cloudfront_distribution",
20       "name": "cdn",
21       "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
22       "instances": [
23         {
24           "schema_version": 1,
25           "attributes": {
26             "aliases": [],
27             "arn": "arn:aws:cloudfront::481111845198:distribution/E3SPHGCQ8JZG6Z",
28             "caller_reference": "terraform-20250601112917055600000001",
29             "comment": null,
30             "continuous_deployment_policy_id": "",
31             "custom_error_response": [],
32             "default_cache_behavior": [
33               {
34                 "allowed_methods": [
35                   "GET",
36                   "HEAD"
37                 ],
38                 "cache_policy_id": "",
```

```
39         "cached_methods": [
40           "GET",
41           "HEAD"
42         ],
43         "compress": false,
44         "default_ttl": 0,
45         "field_level_encryption_id": "",
46         "forwarded_values": [
47           {
48             "cookies": [
49               {
50                 "forward": "none",
51                 "whitelisted_names": []
52               }
53             ],
54             "headers": [],
55             "query_string": false,
56             "query_string_cache_keys": []
57           }
58         ],
59         "function_association": [],
60         "grpc_config": [
61           {
62             "enabled": false
63           }
64         ],
65         "lambda_function_association": [],
66         "max_ttl": 0,
67         "min_ttl": 0,
68         "origin_request_policy_id": "",
69         "realtime_log_config_arn": "",
70         "response_headers_policy_id": "",
71         "smooth_streaming": false,
72         "target_origin_id": "S3Origin",
73         "trusted_key_groups": [],
74         "trusted_signers": [],
75         "viewer_protocol_policy": "redirect-to-https"
76       }
77     ],
78     "default_root_object": "index.html",
79     "domain_name": "dm7dyutyusru4.cloudfront.net",
80     "enabled": true,
81     "etag": "E3VILL98WLYTQ2",
```

Main.tf Script Documentation / Code Snippets

The **primary configuration file** in my project.

This is where I have defined the resources I'm creating and managing.

```
1  provider "aws" {
2    region = var.region
3  }
4
5  # S3 bucket with a unique name
6  resource "aws_s3_bucket" "website" {
7    bucket = "${var.project_name}-bucket-kennedy-2025"
8
9    tags = {
10      Project = var.project_name
11    }
12  }
13
14  # Website configuration
15  resource "aws_s3_bucket_website_configuration" "website_config" {
16    bucket = aws_s3_bucket.website.id
17
18    index_document {
19      suffix = "index.html"
20    }
21  }
22
23  # Allow public access
24  resource "aws_s3_bucket_public_access_block" "block" {
25    bucket = aws_s3_bucket.website.id
26
27    block_public_acls       = false
28    block_public_policy     = false
29    ignore_public_acls     = false
30    restrict_public_buckets = false
31  }
32
33  # Attach public read policy to allow HTML page access
34  resource "aws_s3_bucket_policy" "public_read" {
35    bucket = aws_s3_bucket.website.id
36
37    policy = jsonencode({
38      Version = "2012-10-17",
39      Statement = [
40        {
41          Sid      = "PublicReadGetObject",
```

```
42          Effect   = "Allow",
43          Principal = "*",
44          Action    = "s3:GetObject",
45          Resource  = "${aws_s3_bucket.website.arn}/*"
46        }
47      ]
48    })
49  }
50
51  # Upload index.html to the bucket
52  resource "aws_s3_object" "html" {
53    bucket      = aws_s3_bucket.website.id
54    key         = "index.html"
55    source      = "${path.module}/index.html"
56    content_type = "text/html"
57  }
58
59  resource "aws_cloudfront_origin_access_control" "oac" {
60    name                = "${var.project_name}-oac"
61    description         = "Access control for CloudFront to S3"
62    origin_access_control_origin_type = "s3"
63    signing_behavior    = "always"
64    signing_protocol    = "sigv4"
65  }
66
67  resource "aws_cloudfront_distribution" "cdn" {
68    enabled              = true
69    default_root_object = "index.html"
70
71    origin {
72      domain_name = aws_s3_bucket.website.bucket_regional_domain_name
73      origin_id   = "S3Origin"
74
75      origin_access_control_id = aws_cloudfront_origin_access_control.oac.id
76    }
77
78    default_cache_behavior {
79      allowed_methods = ["GET", "HEAD"]
80      cached_methods  = ["GET", "HEAD"]
81      target_origin_id = "S3Origin"
82
83      viewer_protocol_policy = "redirect-to-https"
84
85      forwarded_values {
```

variables.tf Script Documentation / Code Snippets

This is where I have **declared my input variables** for my Terraform configuration.

```
1  variable "region" {
2      default = "eu-central-1"
3  }
4
5  variable "project_name" {
6      default = "static-website"
7  }
```

outputs.tf Script Documentation / Code Snippets

This is where I have **defined my output values** from my Terraform configuration:

The output shows the crucial information needed to access or verify my deployed website, such as:

- Website Endpoint URL:** The URL provided by the cloud provider for my website.
- CDN Domain Name:** its domain name.
- Bucket ARN:** The Amazon Resource Name (ARN) of the S3 bucket.

```
1  output "website_url" {
2      value = "http://${aws_s3_bucket.website.bucket}.s3-website.${var.region}.amazonaws.com"
3  }
4  output "cloudfront_url" {
5      value = "https://${aws_cloudfront_distribution.cdn.domain_name}"
6  }
```


Index.html Script Documentation / Code Snippets

This foundational HTML document defines the structure and initial content of my website's main page.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Cloud Programming Project</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             margin: 2rem;
11             background: #f9f9f9;
12         }
13         h1 {
14             color: #2c3e50;
15         }
16         code {
17             background-color: #e6e6e6;
18             padding: 2px 4px;
19             border-radius: 3px;
20         }
21     </style>
22 </head>
23 <body>
24 <h1>Cloud Programming Project – Static Website Deployment</h1>
25 <p>This website was deployed using <strong>AWS S3, CloudFront, and Terraform</strong> as part of the DLB
26
```

```
27 <h2>✅ What This Project Demonstrates</h2>
28 <ul>
29     <li>Infrastructure-as-Code (IaC) using <code>Terraform</code></li>
30     <li>Hosting a static website on <code>AWS S3</code></li>
31     <li>Global content delivery via <code>AWS CloudFront</code></li>
32     <li>Secure HTTPS delivery using <code>CloudFront default SSL</code></li>
33 </ul>
34
35 <h2>🌐 Technologies Used</h2>
36 <ul>
37     <li>AWS S3 (for static file storage)</li>
38     <li>AWS CloudFront (for global CDN & HTTPS)</li>
39     <li>Terraform (for full environment provisioning)</li>
40 </ul>
41
42 <h2>📁 Project Files</h2>
43 <ul>
44     <li><code>main.tf</code> – Infrastructure definitions</li>
45     <li><code>variables.tf</code> – AWS region and project name</li>
46     <li><code>outputs.tf</code> – Display URLs after deployment</li>
47     <li><code>index.html</code> – This web page</li>
48 </ul>
49
50 <p>Built with ❤️ for IU Cloud Programming Portfolio (Task 1)</p>
51 </body>
52 </html>
```

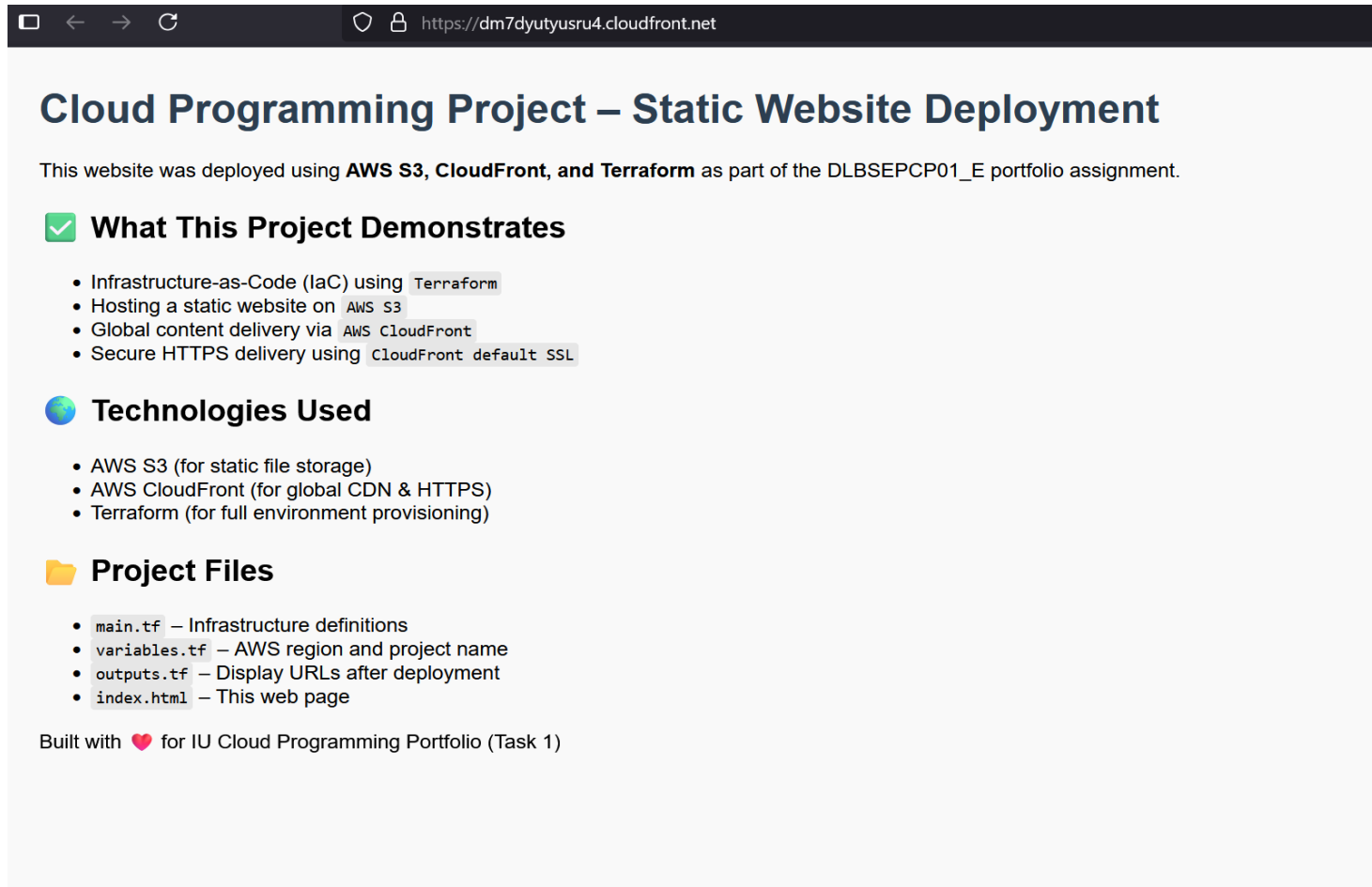
Conclusion

Access my Website:

After a successful deployment, you'll see an output like that shown below

Click the URL or paste it into a browser to view the hosted website!

website_url = <http://static-website-bucket-kennedy-2025.s3-website.eu-central-1.amazonaws.com>



✅ What This Project Demonstrates

- Infrastructure-as-Code (IaC) using Terraform
- Hosting a static website on AWS S3
- Global content delivery via AWS CloudFront
- Secure HTTPS delivery using CloudFront default SSL

🌐 Technologies Used

- AWS S3 (for static file storage)
- AWS CloudFront (for global CDN & HTTPS)
- Terraform (for full environment provisioning)

📁 Project Files

- main.tf – Infrastructure definitions
- variables.tf – AWS region and project name
- outputs.tf – Display URLs after deployment
- index.html – This web page

Built with ❤️ for IU Cloud Programming Portfolio (Task 1)