

# Homework #4 : Automate Deployment

Team Name: Nano Clouds

Team Members: Apoorva, Junie, Rinku, Ananya

Date: 04/10/2025

---

**Purpose:** To automate the build and deployment process from a code repository using configuration automation, CI/CD tools/plugins, and a cloud-based CI/CD service.

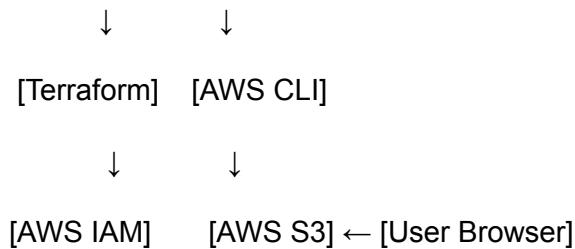
## Technologies Used:

- Version Control: GitHub
  - Cloud Provider: AWS (S3, IAM)
  - CI/CD Service: GitHub Actions
  - Infrastructure as Code (IaC): Terraform
  - Configuration Automation / Deployment Tool: AWS CLI (via aws s3 sync)
- 

System overview:

## Architecture diagram :

[Developer PC] → [GitHub Repo] → [GitHub Actions]



## Integration Design:

- **Code Repository (GitHub):**
  - <https://github.com/ananya101001/Automate-Deployment>
  - index.html in root, infrastructure/ folder for Terraform, .github/workflows/ for Actions
  - The pipeline is configured to trigger automatically upon pushes to the main branch is shown in the below screenshot.
- **Cloud Provider (AWS):**
  - AWS S3 for static website hosting:  
<http://my-static-site-ananya-20250410.s3-website-us-east-1.amazonaws.com/>

- AWS IAM is used to provide secure, programmatic credentials (via Access Keys stored and secrets stored as GitHub Secrets) for the pipeline to interact with AWS.
- **CI/CD Service (GitHub Actions):**
  - on: push: branches: [main]
  - jobs: a single deploy job
  - GitHub Actions used (plugins): actions/checkout@v4, aws-actions/configure-aws-credentials@v4, hashicorp/setup-terraform@v3.
  - Secrets (AWS\_ACCESS\_KEY\_ID, AWS\_SECRET\_ACCESS\_KEY) are securely passed from GitHub settings to the configure-aws-credentials action.
- **Infrastructure as Code (Terraform):**
  - To define and manage the AWS S3 bucket infrastructure declaratively.
  - The resources managed by Terraform (e.g., aws\_s3\_bucket, aws\_s3\_bucket\_website\_configuration, aws\_s3\_bucket\_policy, aws\_s3\_bucket\_public\_access\_block).
  - Integration: The pipeline runs terraform init, terraform validate, and terraform apply -auto-approve.
- **Configuration Automation (AWS CLI):**
  - Deploying website content to S3, the "configuration automation" step involves ensuring the correct files are present in the bucket.
  - The aws s3 sync  
`.https://us-east-1.console.aws.amazon.com/s3/buckets/my-static-site-ananya-20250410?region=us-east-1&bucketType=general&tab=objects --delete` command within the pipeline automates this configuration process. The --delete flag ensures the S3 content exactly mirrors the repository's state.
  - Run as a script step in the GitHub Actions workflow after Terraform provisions the bucket.

## Configuration Details

- **GitHub Actions Workflow (.github/workflows/main.yml):**

```

.github > workflows > ! main.yml
  1 name: Deploy Static Site to S3
  2
  3 on:
  4   push:
  5     branches:
  6       - main # Trigger deployment only on pushes to the main branch
  7
  8 jobs:
  9   deploy:
10     runs-on: ubuntu-latest
11     # Granting permissions for fetching OIDC token and assuming role (optional but good practice)
12     # If not using OIDC, you'll rely solely on the AWS_ACCESS_KEY_ID/SECRET_ACCESS_KEY secrets
13     permissions:
14       id-token: write # required for OIDC if you use it
15       contents: read # required to checkout the code
16
17 steps:
18   - name: Checkout Code
19     uses: actions/checkout@v4
20
21   - name: Configure AWS Credentials
22     uses: aws-actions/configure-aws-credentials@v4
23     with:
24       aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
25       aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
26       aws-region: us-east-1 # Must match the region in your terraform file
27
28   - name: Setup Terraform
29     uses: hashicorp/setup-terraform@v3
30     # Optionally specify terraform_version if needed
31
32   - name: Terraform Init
33     id: init
34     run: terraform init
35     working-directory: ./infrastructure # Run in the infrastructure subfolder
36
37   - name: Terraform Validate
38     id: validate

```

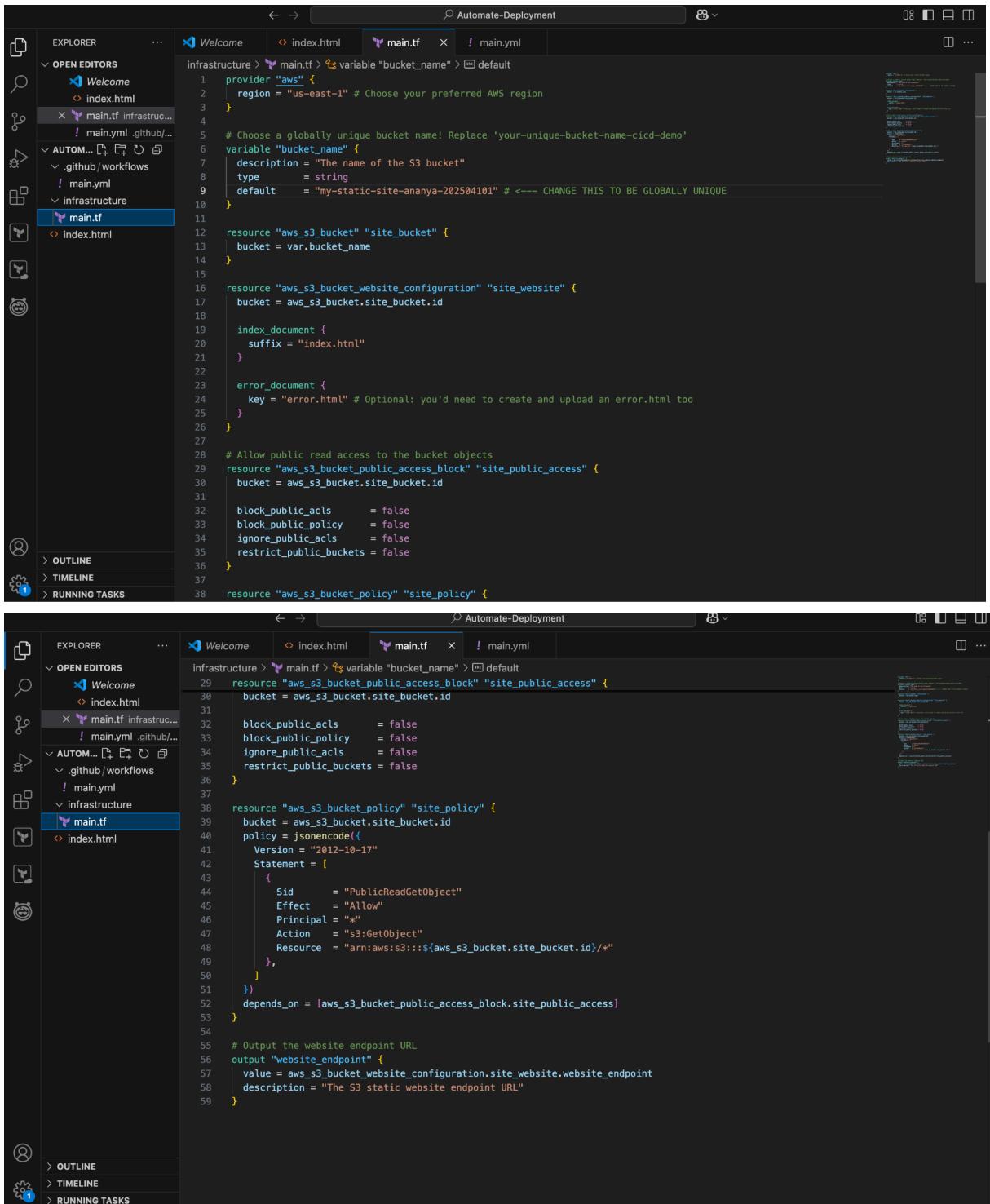
```

.github > workflows > ! main.yml
  8 jobs:
  9   deploy:
10     steps:
11       - name: Terraform Validate
12
13       - name: Terraform Apply
14         id: apply
15         run: terraform apply --auto-approve --no-color
16         working-directory: ./infrastructure
17
18       - name: Deploy site files to S3
19         run: |
20           # Replace with your unique bucket name used in main.tf
21           BUCKET_NAME="my-static-site-ananya-20250410" # <-- SAME UNIQUE NAME AS IN TERRAFORM
22           # Sync files - assuming index.html is in the root. If in src/, change '.' to './src'
23           aws s3 sync . s3://${BUCKET_NAME}/ --delete
24
25           # working-directory: ./src # Uncomment and adjust sync path if your site files are in src/
26
27       - name: Update Job Summary
28         run: |
29             echo "## Deployment Summary" >> $GITHUB_STEP_SUMMARY
30             echo "- **Terraform Apply:** Success" >> $GITHUB_STEP_SUMMARY
31             echo "- **S3 Sync:** Success" >> $GITHUB_STEP_SUMMARY
32             WEBSITE_URL=$(cd ./infrastructure && terraform output -raw website_endpoint)
33             echo "- **Website URL:** http://${WEBSITE_URL}" >> $GITHUB_STEP_SUMMARY
34             echo "[Visit Deployed Site](http://${WEBSITE_URL})" >> $GITHUB_STEP_SUMMARY
35
36       - name: Output Website URL
37         run: |
38             # Extract the URL from Terraform output (simple approach)
39             WEBSITE_URL=$(terraform output -raw website_endpoint)
40             echo "Website deployed to: http://${WEBSITE_URL}"
41
42             working-directory: ./infrastructure

```

- **Terraform Configuration (infrastructure/main.tf):**

● **AWS IAM Configuration:**



```

Welcome          index.html      main.tf      main.yml
infrastructure > main.tf > variable "bucket_name" > default
1 provider "aws" {
2   region = "us-east-1" # Choose your preferred AWS region
3 }
4
5 # Choose a globally unique bucket name! Replace 'your-unique-bucket-name-cicd-demo'
6 variable "bucket_name" {
7   description = "The name of the S3 bucket"
8   type        = string
9   default     = "my-static-site-ananya-202504101" # <--- CHANGE THIS TO BE GLOBALLY UNIQUE
10 }

11 resource "aws_s3_bucket" "site_bucket" {
12   bucket = var.bucket_name
13 }
14

15 resource "aws_s3_bucket_website_configuration" "site_website" {
16   bucket = aws_s3_bucket.site_bucket.id
17
18   index_document {
19     suffix = "index.html"
20   }
21
22   error_document {
23     key = "error.html" # Optional: you'd need to create and upload an error.html too
24   }
25 }
26

27 # Allow public read access to the bucket objects
28 resource "aws_s3_bucket_public_access_block" "site_public_access" {
29   bucket = aws_s3_bucket.site_bucket.id
30
31   block_public_acls      = false
32   block_public_policy    = false
33   ignore_public_acls    = false
34   restrict_public_buckets = false
35 }
36

37 resource "aws_s3_bucket_policy" "site_policy" {
38   bucket = aws_s3_bucket.site_bucket.id
39   policy = jsonencode({
40     Version = "2012-10-17"
41     Statement = [
42       {
43         Sid      = "PublicReadGetObject"
44         Effect   = "Allow"
45         Principal = "*"
46         Action   = "s3:GetObject"
47         Resource  = "arn:aws:s3:::${aws_s3_bucket.site_bucket.id}/*"
48       },
49     ]
50   })
51   depends_on = [aws_s3_bucket_public_access_block.site_public_access]
52 }
53

54 # Output the website endpoint URL
55 output "website_endpoint" {
56   value = aws_s3_bucket_website_configuration.site_website.website_endpoint
57   description = "The S3 static website endpoint URL"
58 }
59

```

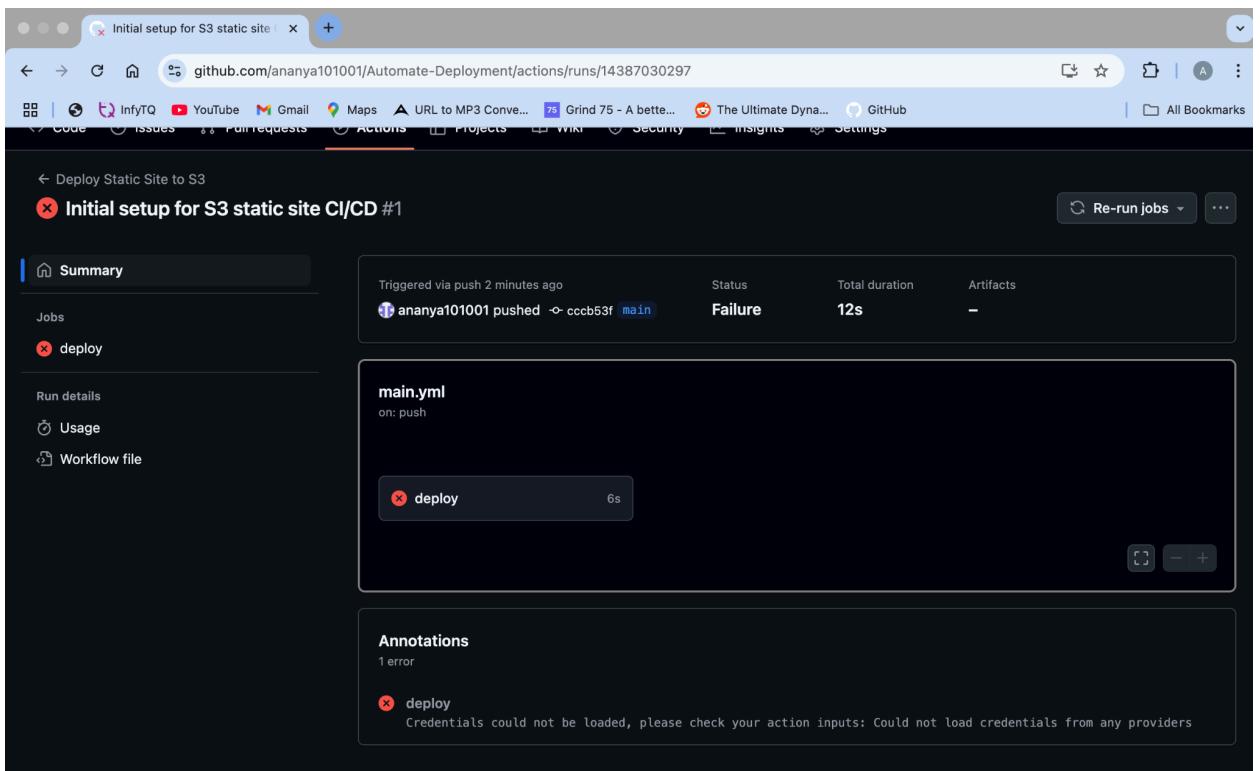
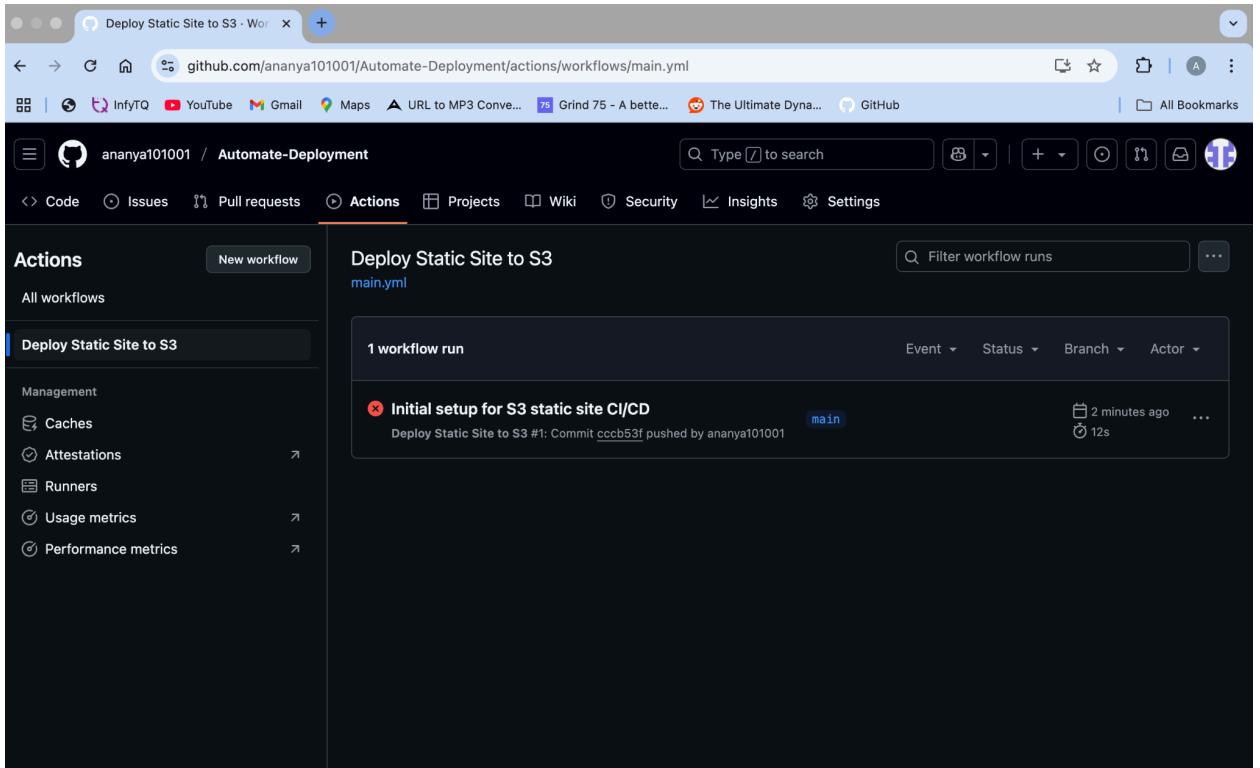
The screenshot shows the GitHub Settings page for the repository 'ananya101001/Automate-Deployment'. The left sidebar is collapsed, and the main area displays 'Environment secrets' and 'Repository secrets'.  
**Environment secrets:**  
Message: 'This environment has no secrets.'  
Action: 'Manage environment secrets'  
**Repository secrets:**  
Title: 'Repository secrets' with a 'New repository secret' button.  
Table:  
Name	Last updated
AWS\_ACCESS\_KEY\_ID	1 hour ago
AWS\_SECRET\_ACCESS\_KEY	1 hour ago

## Screenshots Demonstration:

### 1. Deploying the first commit

The screenshot shows the GitHub Actions interface for the workflow 'Initial setup for S3 static site CI/CD #1'.  
Workflow status: 'In progress'  
Workflow file: 'main.yml'  
Trigger: 'on: push'  
Job: 'deploy' (status: 'In progress', duration: 5s)  
Actions menu: 'Cancel workflow', 'Latest #2', '...'.

### 2. Failed due to No secret keys



3. Added secret key and pushed to main again

The screenshot shows the GitHub Actions settings page for a repository. On the left, there's a sidebar with options like Rules, Actions, Webhooks, Environments, Codespaces, and Pages. Under Security, there are Code Security, Deploy keys, and Secrets and variables. The Secrets and variables section is expanded, showing two repository secrets: AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY, both updated 1 hour ago.

#### 4. Again Failed due to improper bucket name

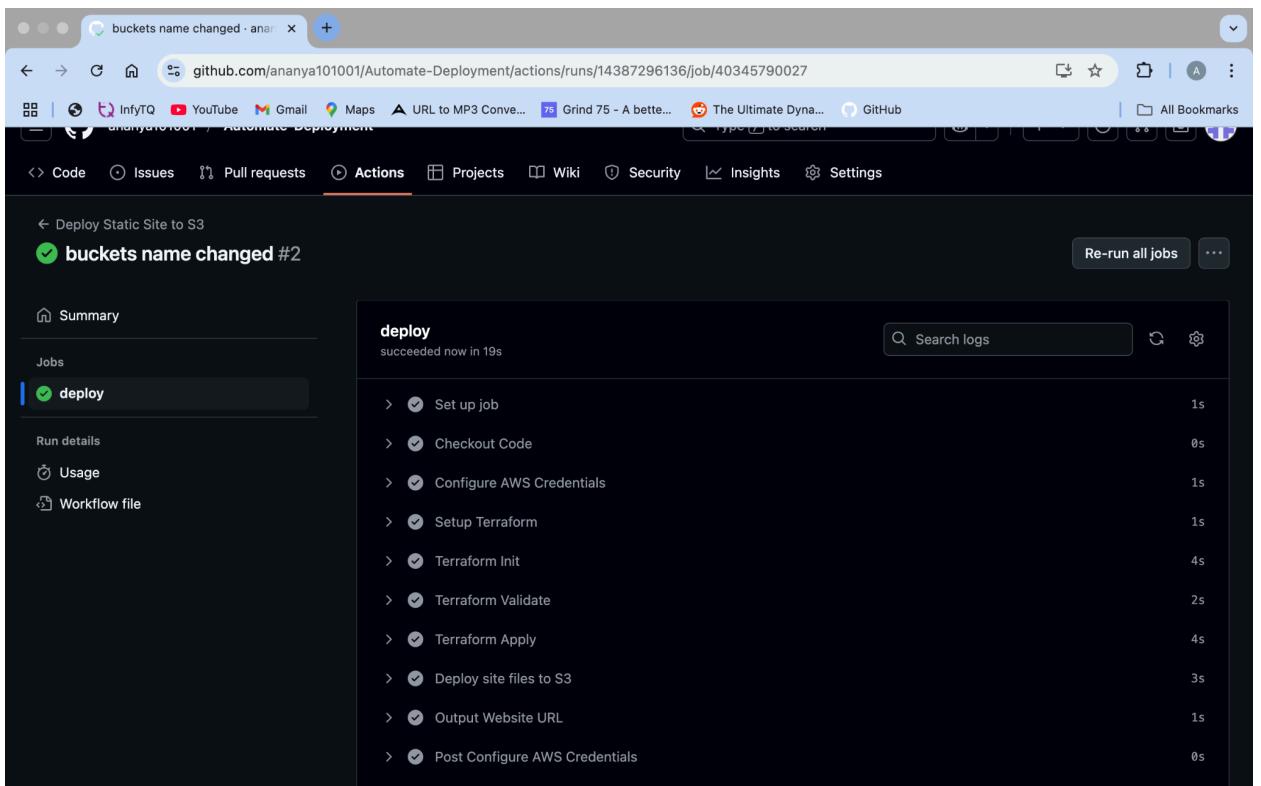
The screenshot shows the GitHub Actions run logs for a job named "deploy". The job failed now in 15s. The logs show a Terraform validation step followed by a Terraform apply step. The apply step shows the following output:

```
1 ► Run terraform apply -auto-approve -no-color
10
11 Terraform used the selected providers to generate the following execution
12 plan. Resource actions are indicated with the following symbols:
13   + create
14
15 Terraform will perform the following actions:
16
17 # aws_s3_bucket.site_bucket will be created
18 + resource "aws_s3_bucket" "site_bucket" {
19   + acceleration_status      = (known after apply)
20   + acl                      = (known after apply)
21   + arn                      = (known after apply)
22   + bucket                   = "my_bucket"
23   + bucket_domain_name       = (known after apply)
24   + bucket_prefix             = (known after apply)
25   + bucketRegionalDomainName = (known after apply)
26   + force_destroy             = false
27   + hosted_zone_id           = (known after apply)
28   + id                       = (known after apply)
29   + object_lock_enabled       = (known after apply)
30   + policy                   = (known after apply)
31   + region                   = (known after apply)
32   + request_payer             = (known after apply)
33   + tags_all                 = (known after apply)
```

#### 5. Changed bucket name and pushed to main again

```
nothing to commit, working tree clean
● akshashe@Akshays-MacBook-Air Automate-Deployment % git push origin main
  Everything up-to-date
● akshashe@Akshays-MacBook-Air Automate-Deployment % git add .
● akshashe@Akshays-MacBook-Air Automate-Deployment % git commit -m "buckets name changed"
[main 047d1f7] buckets name changed
  2 files changed, 2 insertions(+), 2 deletions(-)
● akshashe@Akshays-MacBook-Air Automate-Deployment % git push origin main
  Enumerating objects: 13, done.
  Counting objects: 100% (13/13), done.
  Delta compression using up to 8 threads
  Compressing objects: 100% (4/4), done.
  Writing objects: 100% (7/7), 596 bytes | 596.00 KiB/s, done.
  Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
  remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
  To https://github.com/ananya101001/Automate-Deployment.git
    cccb53f..047d1f7  main -> main
○ akshashe@Akshays-MacBook-Air Automate-Deployment %
```

## 6. Pipeline successfully ran



## 7. Buckets seen in AWS console

The screenshot shows the AWS S3 console with the URL [us-east-1.console.aws.amazon.com/s3/home?region=us-east-1#](https://us-east-1.console.aws.amazon.com/s3/home?region=us-east-1#). The left sidebar includes links for General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings, Storage Lens (Dashboards, Storage Lens groups), and AWS Organizations settings. The main content area displays an 'Account snapshot - updated every 24 hours' section with a link to 'View Storage Lens dashboard'. Below it, the 'General purpose buckets' tab is selected, showing a table with one item: 'my-static-site-ananya-20250410'. The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date.

Name	AWS Region	IAM Access Analyzer	Creation date
my-static-site-ananya-20250410	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	April 10, 2025, 11:07:50 (UTC-07:00)

The screenshot shows the AWS S3 console with the URL [us-east-1.console.aws.amazon.com/s3/buckets/my-static-site-ananya-20250410?region=us-east-1&bucketType=general&tab=objects](https://us-east-1.console.aws.amazon.com/s3/buckets/my-static-site-ananya-20250410?region=us-east-1&bucketType=general&tab=objects). The left sidebar is identical to the previous screenshot. The main content area shows the 'my-static-site-ananya-20250410' bucket with the 'Objects' tab selected. It displays four objects: '.git/' (Folder), '.github/' (Folder), 'index.html' (html), and 'infrastructure/' (Folder). The table columns are Name, Type, Last modified, Size, and Storage class.

Name	Type	Last modified	Size	Storage class
.git/	Folder	-	-	-
.github/	Folder	-	-	-
index.html	html	April 10, 2025, 11:07:52 (UTC-07:00)	418.0 B	Standard
infrastructure/	Folder	-	-	-

## 8. Got website URL

The screenshot shows a GitHub Actions run details page for a deployment job. The job status is "succeeded 4 minutes ago in 19s". The job steps listed are:

- > ✓ Set up job 1s
- > ✓ Checkout Code 0s
- > ✓ Configure AWS Credentials 1s
- > ✓ Setup Terraform 1s
- > ✓ Terraform Init 4s
- > ✓ Terraform Validate 2s
- > ✓ Terraform Apply 4s
- > ✓ Deploy site files to S3 3s

A specific step, "Output Website URL", is highlighted with a blue border. The output for this step is:

```
1 ► Run # Extract the URL from Terraform output (simple approach)
12 Website deployed to: http://my-static-site-ananya-20250410.s3-website-us-east-1.amazonaws.com
```

Following this, the remaining steps are:

- > ✓ Post Configure AWS Credentials 0s
- > ✓ Post Checkout Code 0s
- > ✓ Complete job 0s

## 9. Displaying website in browser

The screenshot shows a web browser window with two tabs: "buckets name changed - ana" and "My Simple CI/CD Website". The active tab is "My Simple CI/CD Website" and displays the URL "my-static-site-ananya-20250410.s3-website-us-east-1.amazonaws.com". The page content is:

# Hello from CI/CD!

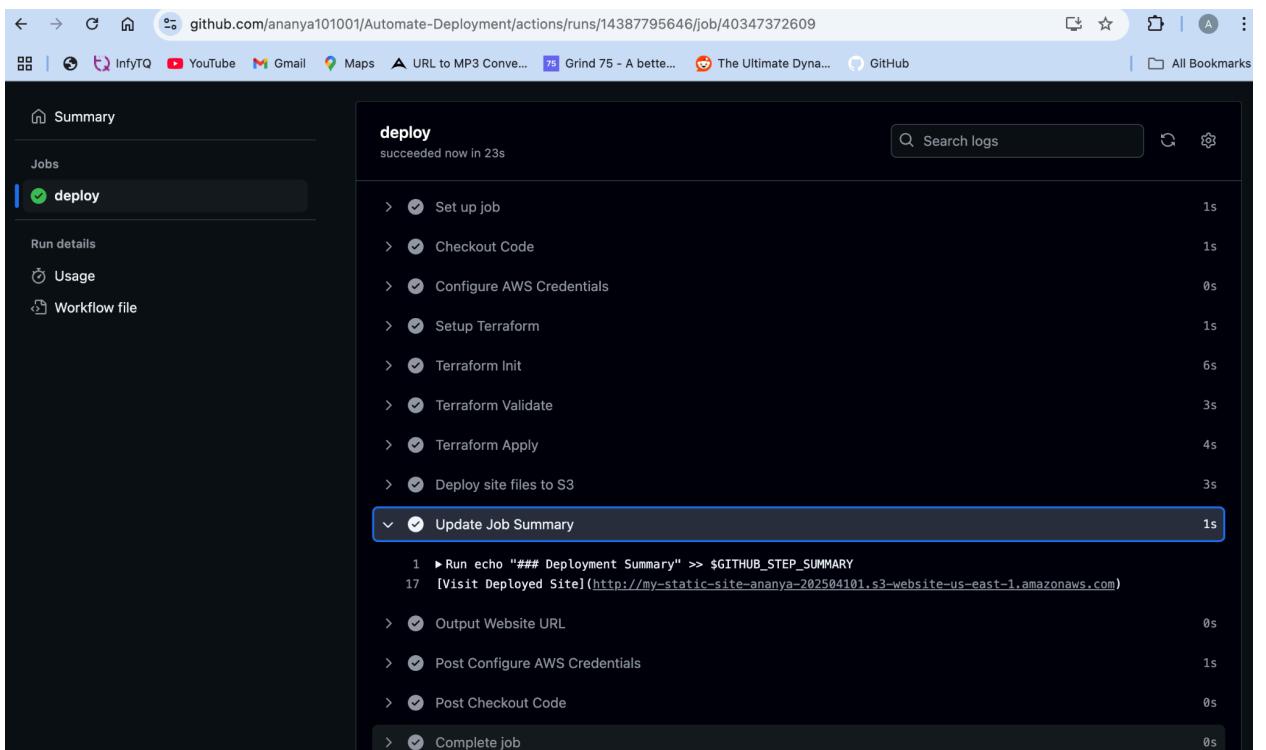
This website was deployed automatically using GitHub Actions and Terraform to AWS S3.

Deployment time: Thu, 10 Apr 2025 18:12:09 GMT

## 10. Updated the pipeline code to show the summary

```
54      - name: Update Job Summary
55        run: |
56          echo "### Deployment Summary" >> $GITHUB_STEP_SUMMARY
57          echo "- **Terraform Apply:** Success" >> $GITHUB_STEP_SUMMARY
58          echo "- **S3 Sync:** Success" >> $GITHUB_STEP_SUMMARY
59          WEBSITE_URL=$(cd ./infrastructure && terraform output -raw website_endpoint)
60          echo "- **Website URL:** http://$WEBSITE_URL" >> $GITHUB_STEP_SUMMARY
61          echo "" >> $GITHUB_STEP_SUMMARY # Add a newline
62          echo "[Visit Deployed Site](http://$WEBSITE_URL)"
63          echo "[Visit Deployed Site](http://$WEBSITE_URL)" >> $GITHUB_STEP_SUMMARY
64
65
66      - name: Output Website URL
67        run: |
68          # Extract the URL from Terraform output (simple approach)
69          WEBSITE_URL=$(terraform output -raw website_endpoint)
70          echo "Website deployed to: http://$WEBSITE_URL"
71
71  working-directory: ./infrastructure
```

## 11. Deploying new pipeline



## 12. Github repo with Deployment pipeline

The screenshot shows the GitHub repository page for 'Automate-Deployment'. The repository was created by 'ananya101001' and has 4 commits. The commits are:

- ananya101001 updated job summary pipeline and changed bucket name
- .github/workflows updated job summary pipeline and changed bucket name
- infrastructure updated job summary pipeline and changed bucket name
- index.html Initial setup for S3 static site CI/CD

The repository has 1 branch and 0 tags. It has 1 watch and 0 forks. There are no releases published.

The screenshot shows a GitHub Actions check status. It displays the message "All checks have passed" and "1 successful check". One check is listed: "Deploy Static Site to S3 / deploy (push)" which was successful in 23s.

13.

14. Added Destroy.yml to destroy all the resources created in AWS

The screenshot shows the GitHub Actions page for the 'Automate-Deployment' repository. The left sidebar shows actions like 'Deploy Static Site to S3' and 'Destroy Infrastructure'. The main area shows 'All workflows' and 'All workflow runs'.

**All workflows**

**All workflow runs**

Event	Status	Branch	Actor
Deploy Static Site to S3 #5: Commit 747729b pushed by ananya101001	main	2 minutes ago	...
Deploy Static Site to S3 #4: Commit 1fa3d80 pushed by ananya101001	main	51 minutes ago	...
Deploy Static Site to S3 #3: Commit 591470c pushed by ananya101001	main	1 hour ago	...
Deploy Static Site to S3 #2: Commit 047df1f7 pushed by ananya101001	main	1 hour ago	...
Deploy Static Site to S3 #1: Commit cccb53f pushed by ananya101001	main	1 hour ago	...

15. If i click on Destroy Infrastructure the workflow will run to destroy all

The screenshot shows the GitHub Actions interface for a workflow named 'Destroy Infrastructure'. The left sidebar lists other workflows like 'Deploy Static Site to S3'. The main panel displays the workflow configuration file 'destroy.yml' with the instruction 'This workflow has a workflow\_dispatch event trigger.' A large button labeled 'Run workflow' is visible. Below it, a message states 'This workflow has no runs yet.' with a play icon.

16. Infrastructure destroying initiated

The screenshot shows the GitHub Actions interface after initiating a workflow run. A message at the top says 'Workflow run was successfully requested.' The main panel shows the 'Destroy Infrastructure' workflow with one run listed: 'Destroy Infrastructure #1: Manually run by ananya101001' on the 'main' branch, which is currently 'In progress'.

17. Done destroying

The screenshot shows a GitHub Actions workflow named "Destroy Infrastructure" in a repository. The workflow has a single run, which was manually triggered by the user. The run completed successfully 16 seconds ago. The workflow has a "Workflow\_dispatch" event trigger.

**Actions**

- New workflow
- All workflows
- Deploy Static Site to S3
- Destroy Infrastructure**
- Management
  - Caches
  - Attestations
  - Runners
  - Usage metrics
  - Performance metrics

**Destroy Infrastructure**  
destroy.yml

0 workflow runs

This workflow has a `workflow_dispatch` event trigger.

**Destroy Infrastructure** main now 16s

---

Github repo link : <https://github.com/ananya101001/Automate-Deployment>