# Zintora Internship Project - Static Website Deployment with CI/CD

This project was completed as part of my internship with Zintora. It demonstrates the deployment of a static website to AWS S3 using Terraform, Bash scripting, and GitHub Actions for a fully automated CI/CD pipeline.

## Project Details

• \*\*GitHub Repository\*\*: https://github.com/Divyam0017/static-website  
• \*\*AWS S3 Bucket Name\*\*: zintora-static-site  
• \*\*Technologies Used\*\*: AWS S3, AWS IAM, Terraform, GitHub Actions, Bash, Linux (WSL)  
• \*\*Operating System\*\*: Ubuntu on WSL

## Project Structure

```  
static-website/  
├── index.html # Static website file  
├── terraform/  
│ ├── main.tf # S3 bucket configuration  
│ ├── provider.tf # AWS provider configuration  
│ ├── variables.tf # Variable definitions  
│ ├── outputs.tf # Outputs for bucket URL  
│ └── .gitignore # Ignore terraform state files  
└── .github/workflows/  
 └── deploy.yml # GitHub Actions workflow for deployment  
```

## Steps to Run Locally

1. \*\*Clone the repository\*\*:  
```bash  
git clone https://github.com/Divyam0017/static-website.git  
cd static-website/terraform  
```  
2. \*\*Initialize Terraform\*\*:  
```bash  
terraform init  
```  
3. \*\*Apply the Terraform configuration\*\*:  
```bash  
terraform apply -var="bucket\_name=zintora-static-site"  
```  
4. \*\*Upload the website file manually (if needed)\*\*:  
```bash  
aws s3 cp ../index.html s3://zintora-static-site --website-redirect  
```

## GitHub Actions Workflow

The project includes a `deploy.yml` workflow file that automatically runs Terraform commands and deploys changes to the S3 bucket whenever code is pushed to the `main` branch.  
  
Key features:  
- Configures AWS credentials using GitHub Secrets (`AWS\_ACCESS\_KEY\_ID` and `AWS\_SECRET\_ACCESS\_KEY`)  
- Runs `terraform init` and `terraform apply` for infrastructure changes  
- Uploads updated HTML files to the S3 bucket

## Screenshots

Below screenshots should be included for final submission:  
1. AWS S3 bucket configuration in the console.  
2. Successful GitHub Actions workflow run.  
3. Hosted static website in a browser.

## Conclusion

This project provided hands-on experience in:  
- Setting up AWS S3 for static website hosting  
- Using Terraform for Infrastructure as Code (IaC)  
- Automating deployments via GitHub Actions  
- Managing infrastructure from Linux CLI (WSL) with Bash scripting  
  
It demonstrates my ability to integrate cloud infrastructure with automation tools to create an efficient CI/CD pipeline.