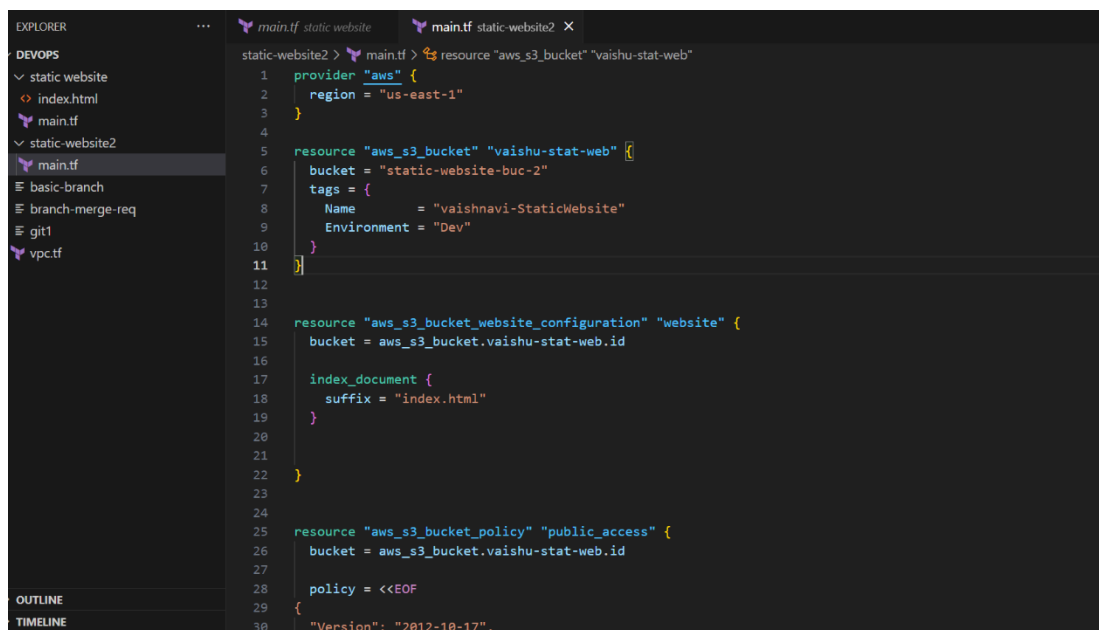


Name: Vaishnavi Padwal

Task: Create an S3 bucket and launch a static website on it using Terraform

Step1: Create terraform configuration file

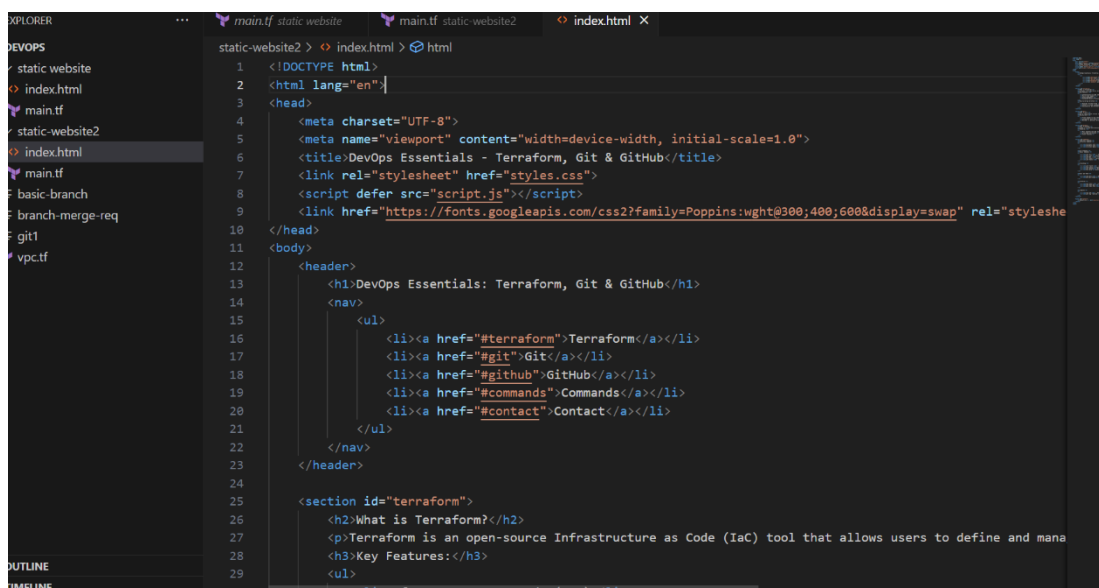
- Go to vs code editor and select folder
- create folder in main selected folder
- create main.tf (configuration file)



```
1 provider "aws" {
2   region = "us-east-1"
3 }
4
5 resource "aws_s3_bucket" "vaishu-stat-web" {
6   bucket = "static-website-buc-2"
7   tags = {
8     Name      = "vaishnavi-StaticWebsite"
9     Environment = "Dev"
10  }
11 }
12
13
14 resource "aws_s3_bucket_website_configuration" "website" {
15   bucket = aws_s3_bucket.vaishu-stat-web.id
16
17   index_document {
18     suffix = "index.html"
19   }
20 }
21
22
23
24
25 resource "aws_s3_bucket_policy" "public_access" {
26   bucket = aws_s3_bucket.vaishu-stat-web.id
27
28   policy = <<EOF
29 {
30   "Version": "2012-10-17",
```

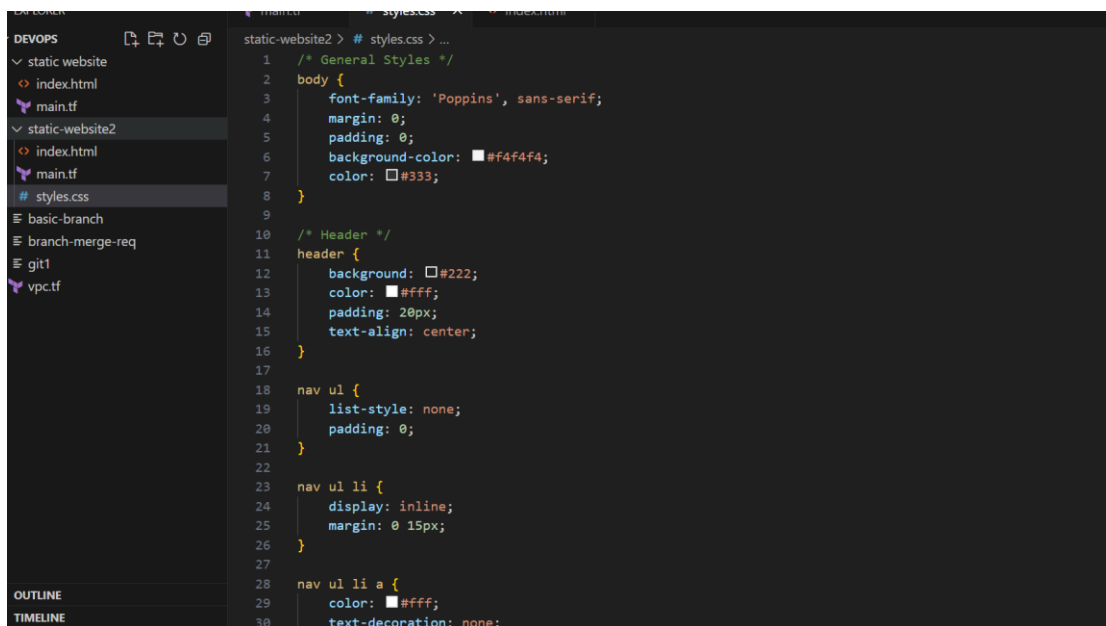
Step 2: index.html (Your Static Website Homepage)

- Create an index.html file in the same directory.



```
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>DevOps Essentials - Terraform, Git & GitHub</title>
7   <link rel="stylesheet" href="styles.css">
8   <script defer src="script.js"></script>
9   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap" rel="style
10 </head>
11 <body>
12   <header>
13     <h1>DevOps Essentials: Terraform, Git & GitHub</h1>
14     <nav>
15       <ul>
16         <li><a href="#terraform">Terraform</a></li>
17         <li><a href="#git">Git</a></li>
18         <li><a href="#github">GitHub</a></li>
19         <li><a href="#commands">Commands</a></li>
20         <li><a href="#contact">Contact</a></li>
21       </ul>
22     </nav>
23   </header>
24
25   <section id="terraform">
26     <h2>What is Terraform?</h2>
27     <p>Terraform is an open-source Infrastructure as Code (IaC) tool that allows users to define and mana
28     <h3>Key Features:</h3>
29     <ul>
30       <li>Infrastructure as Code (IaC)</li>
```

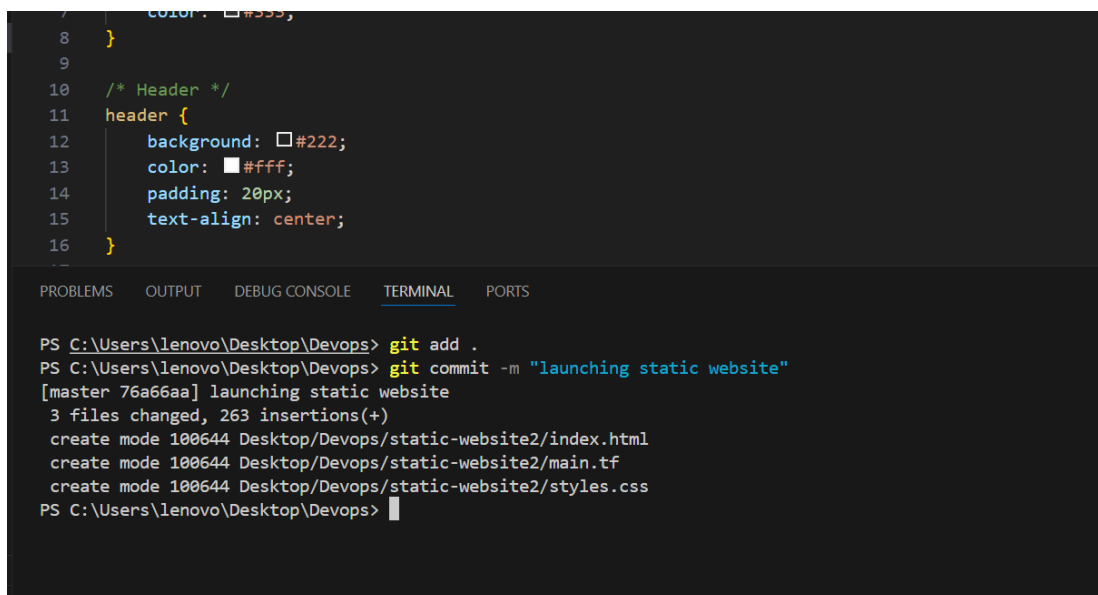
Step 3: create style. file to make website attractive



The screenshot shows the Visual Studio Code editor with a file explorer on the left and a code editor in the center. The file explorer shows a project structure with folders 'static-website' and 'static-website2', and files 'index.html', 'main.tf', and 'styles.css'. The 'styles.css' file is selected and its content is displayed in the editor. The code defines general styles for the body, a header, and a navigation menu.

```
1  /* General Styles */
2  body {
3      font-family: 'Poppins', sans-serif;
4      margin: 0;
5      padding: 0;
6      background-color: #f4f4f4;
7      color: #333;
8  }
9
10 /* Header */
11 header {
12     background: #222;
13     color: #fff;
14     padding: 20px;
15     text-align: center;
16 }
17
18 nav ul {
19     list-style: none;
20     padding: 0;
21 }
22
23 nav ul li {
24     display: inline;
25     margin: 0 15px;
26 }
27
28 nav ul li a {
29     color: #fff;
30     text-decoration: none;
```

Step 4: Now add and commit it to your local repository

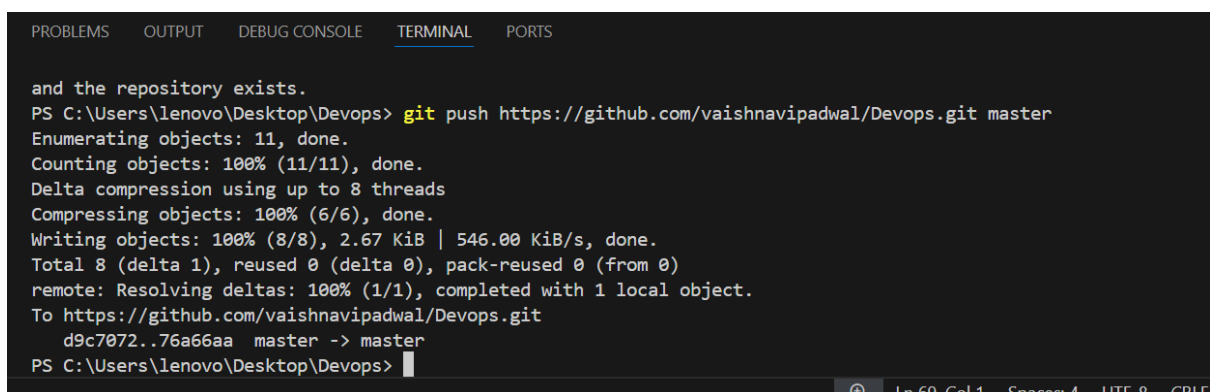


The screenshot shows the Visual Studio Code editor with the 'styles.css' file open. Below the editor, the 'TERMINAL' tab is active, displaying the output of Git commands. The commands executed are 'git add .' and 'git commit -m "launching static website"', which successfully added and committed the files to the local repository.

```
7      color: #333;
8  }
9
10 /* Header */
11 header {
12     background: #222;
13     color: #fff;
14     padding: 20px;
15     text-align: center;
16 }
17
18 nav ul {
19     list-style: none;
20     padding: 0;
21 }
22
23 nav ul li {
24     display: inline;
25     margin: 0 15px;
26 }
27
28 nav ul li a {
29     color: #fff;
30     text-decoration: none;
```

```
PS C:\Users\lenovo\Desktop\Devops> git add .
PS C:\Users\lenovo\Desktop\Devops> git commit -m "launching static website"
[master 76a66aa] launching static website
3 files changed, 263 insertions(+)
create mode 100644 Desktop/Devops/static-website2/index.html
create mode 100644 Desktop/Devops/static-website2/main.tf
create mode 100644 Desktop/Devops/static-website2/styles.css
PS C:\Users\lenovo\Desktop\Devops>
```

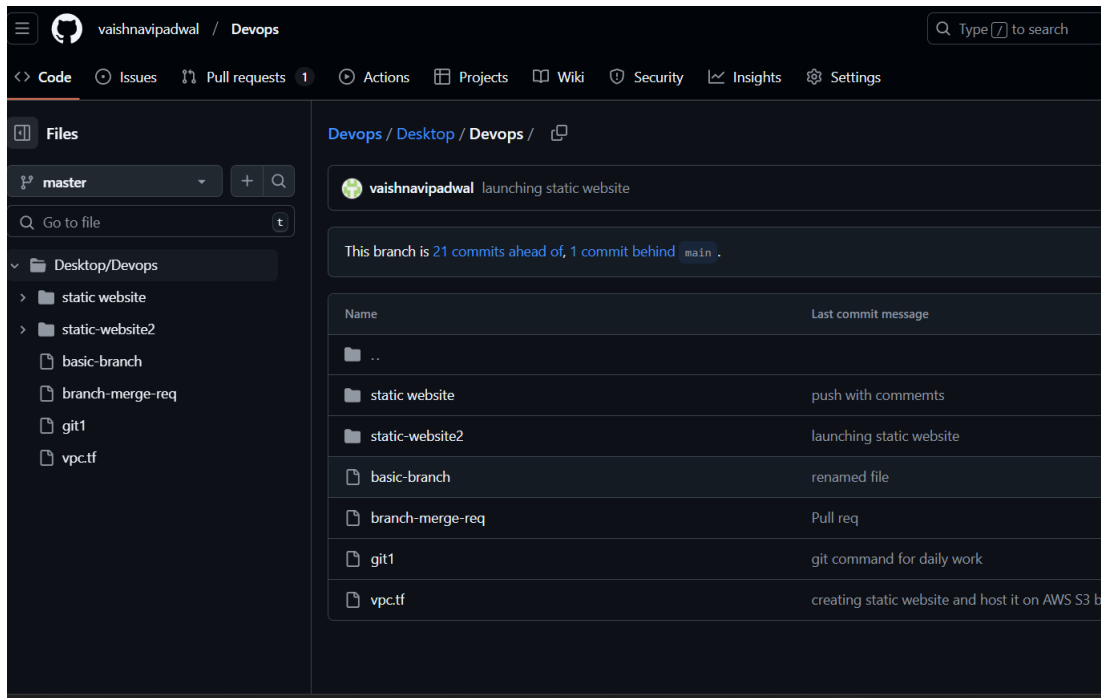
step 5: push it to your remote repository (Github)



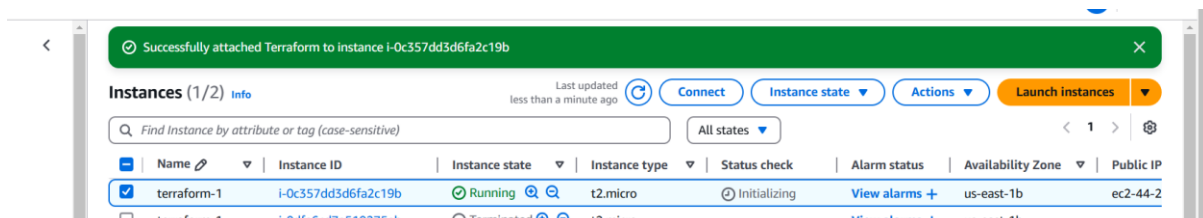
The screenshot shows the Visual Studio Code editor with the 'TERMINAL' tab active. The terminal displays the output of the 'git push' command, which successfully pushed the local changes to the remote repository on GitHub. The output shows the progress of enumerating objects, counting objects, compressing objects, and writing objects, followed by the resolution of deltas and the final push to the master branch.

```
and the repository exists.
PS C:\Users\lenovo\Desktop\Devops> git push https://github.com/vaishnavipadwal/Devops.git master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 2.67 KiB | 546.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/vaishnavipadwal/Devops.git
d9c7072..76a66aa master -> master
PS C:\Users\lenovo\Desktop\Devops>
```

Step 6: Now go to Github and check it whether pushed files is available or not?



Step7: Now go to AWS EC2 resource and create an instance which have full administrative and S3 bucket permissions with valid key-pair



Step 6: Now go to cloud shell and take ssh of that launched instance

```
CloudShell
us-east-1 +
WELCOME TO UBUNTU 24.04.1 LTS (GNU/Linux 5.15.0-102.1-aws aarch64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Thu Feb 13 11:58:09 UTC 2025

System load:  0.03          Processes:            107
Usage of /:   24.9% of 6.71GB Users logged in:          0
Memory usage: 21%          IPv4 address for enx0: 172.31.89.144
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-89-144:~$
```

Step7: login with root user and clone remote repository which have terraform file.

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-89-144:~$ sudo -i
root@ip-172-31-89-144:~# ls
snap
root@ip-172-31-89-144:~# git clone https://github.com/vaishnavipadwal/Devops.git
Cloning into 'Devops'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (45/45), done.
remote: Total 110 (delta 26), reused 100 (delta 20), pack-reused 0 (from 0)
Receiving objects: 100% (110/110), 13.37 KiB | 1.67 MiB/s, done.
Resolving deltas: 100% (26/26), done.
root@ip-172-31-89-144:~#
```

Step8: Now install terraform

```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-89-144:~# terraform --version
Terraform v1.10.5
on linux_amd64
root@ip-172-31-89-144:~#
```

Step9: Go to directory which have terraform file

```
root@ip-172-31-89-144:~# ls
Devops  snap
root@ip-172-31-89-144:~# cd Devops/
root@ip-172-31-89-144:~/Devops# ls
LS: command not found
root@ip-172-31-89-144:~/Devops# ls
README.md
root@ip-172-31-89-144:~/Devops# git checkout master
branch 'master' set up to track 'origin/master'.
Switched to a new branch 'master'
root@ip-172-31-89-144:~/Devops# ls
Desktop
root@ip-172-31-89-144:~/Devops# cd Desktop/
root@ip-172-31-89-144:~/Devops/Desktop# cd Devops/
root@ip-172-31-89-144:~/Devops/Desktop/Devops# ls
basic-branch  branch-merge-req  git1  'static website'  static-website2  vpc.tf
root@ip-172-31-89-144:~/Devops/Desktop/Devops# cd 'static-website2/'
Command 'cs' not found, but can be installed with:
apt install csound
root@ip-172-31-89-144:~/Devops/Desktop/Devops# cd 'static-website2/'
root@ip-172-31-89-144:~/Devops/Desktop/Devops/static-website2#
```

Step10: now run terraform commands step by step:

1. terraform init

```
root@ip-172-31-89-144:~/Devops/Desktop/Devops/static-website2# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.86.1...
- Installed hashicorp/aws v5.86.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-89-144:~/Devops/Desktop/Devops/static-website2#
```

2. terraform plan

```
+ resource "aws_s3_object" "style" {
+   acl                  = (known after apply)
+   arn                  = (known after apply)
+   bucket                = (known after apply)
+   bucket_key_enabled    = (known after apply)
+   checksum_crc32        = (known after apply)
+   checksum_crc32c       = (known after apply)
+   checksum_sha1         = (known after apply)
+   checksum_sha256       = (known after apply)
+   content_type          = "text/css"
+   etag                  = (known after apply)
+   force_destroy         = false
+   id                    = (known after apply)
+   key                   = "style.css"
+   kms_key_id            = (known after apply)
+   server_side_encryption = (known after apply)
+   source                = "style.css"
+   storage_class          = (known after apply)
+   tags_all              = (known after apply)
+   version_id            = (known after apply)
+ }

Plan: 6 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" n
root@ip-172-31-89-144:~/Devops/Desktop/Devops/static-website2#
```

3. terraform apply

```
    + Resource = "arn:aws:s3:::static-website-buc-2/*"
    + Sid      = "PublicReadGetObject"
  },
  + Version = "2012-10-17"
)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_s3_bucket_policy.public_access: Creating...
aws_s3_bucket_policy.public_access: Creation complete after 1s [id=static-website-buc-2]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
s3_static_website_url = "static-website-buc-2.s3-website-us-east-1.amazonaws.com"
root@ip-172-31-89-144:~/Devops/Desktop/Devops/static-website2#
```

Now click the link as output and visit the website

DevOps Essentials: Terraform, Git & GitHub

TerraformGitGitHubCommandsContact

What is Terraform?

Terraform is an open-source Infrastructure as Code (IaC) tool that allows users to define and manage cloud resources using a declarative configuration language.

Key Features:

- Infrastructure as Code (IaC)
- Multi-Cloud Support
- State Management
- Automated Resource Provisioning

How to Set Up Terraform

1. Download Terraform from [official site](#)
2. Install it using a package manager
3. Initialize a new Terraform project using `terraform init`

What is Git?

Git is a distributed version control system that allows multiple developers to work on the same project efficiently.

Why Use Git?

- Track changes to your code
- Collaborate with teams
- Rollback to previous versions

What is GitHub?

GitHub is a cloud-based hosting service for Git repositories, enabling better collaboration and code sharing.

Features:

- Pull requests and code reviews
- Branching and merging
- Issue tracking

Important Commands

Terraform Commands

- `terraform init` - Initialize a new Terraform project
- `terraform plan` - Preview changes
- `terraform apply` - Apply changes

Git Commands

Basic Commands

- `git init` - Initialize a new repository
- `git clone <repository-url>` - Clone an existing repository
- `git add .` - Stage changes
- `git commit -m "message"` - Commit changes
- `git status` - Check current status

Branching

- `git branch` - List branches
- `git branch <branch-name>` - Create a new branch
- `git checkout <branch-name>` - Switch to a branch
- `git merge <branch-name>` - Merge a branch

Pull and Push

- `git pull` - Fetch and merge changes
- `git push origin <branch-name>` - Push changes

Stashing

- `git stash` - Save changes without committing
- `git stash pop` - Apply stashed changes