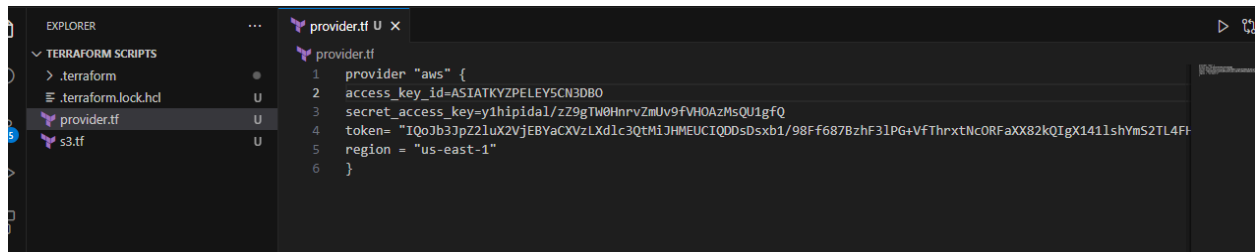
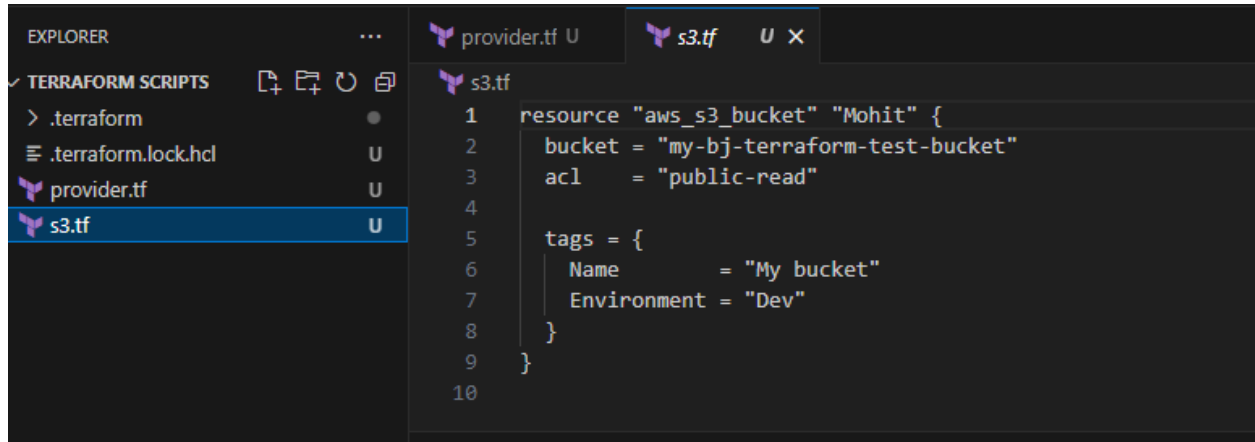


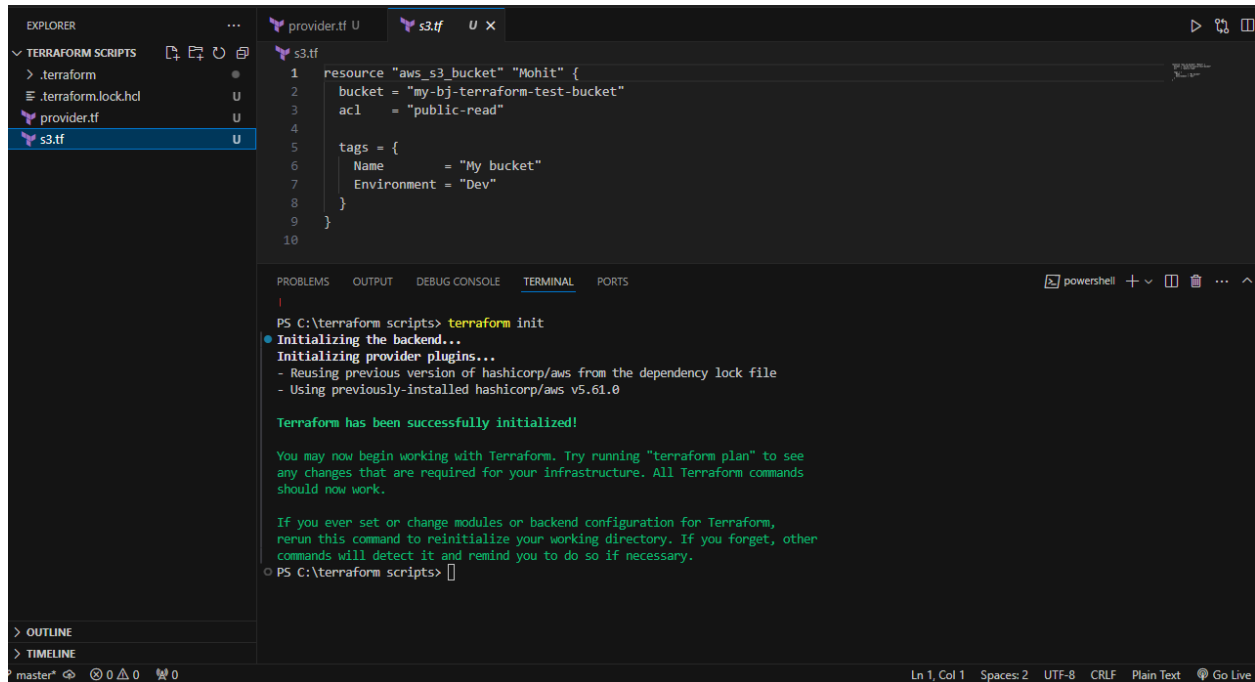
Adv devops 6



```
1 provider "aws" {
2   access_key_id=ASIATKYZPELEY5CN3DB0
3   secret_access_key=y1h1p1dal/zZ9gTw0HnrvZmUv9fVHOAZMsQU1gFQ
4   token= "IQoJb3JpZ2luX2VjEBYCaCXVzLXd1c3QtMjJHMEUCIQD0sDsb1/98Ff687BzhF3lPG+VfThrxNcORFaXX82kQIgX1411shYmS2TL4FI
5   region = "us-east-1"
6 }
```



```
1 resource "aws_s3_bucket" "Mohit" {
2   bucket = "my-bj-terraform-test-bucket"
3   acl    = "public-read"
4
5   tags = {
6     Name       = "My bucket"
7     Environment = "Dev"
8   }
9 }
10
```



```
1 resource "aws_s3_bucket" "Mohit" {
2   bucket = "my-bj-terraform-test-bucket"
3   acl    = "public-read"
4
5   tags = {
6     Name       = "My bucket"
7     Environment = "Dev"
8   }
9 }
10
```

PS C:\terraform scripts> terraform init

- Initializing the backend...
- Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.61.0

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.

PS C:\terraform scripts>

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left displaying the file structure under 'TERRAFORM SCRIPTS'. The file 'provider.tf' is selected and open in the editor. The code in 'provider.tf' defines an AWS provider configuration with the following details:

```
1 provider "aws" {
2   access_key = "ASIATKYZPELEYSCN3DB0"
3   secret_key = "y1hipidal/zZ9gTW0HnrvZmJv9fVHOAzMsQ1gFQ"
4   token = "IQoJb3JpZ2luXZVjEBYCaXVzLXdlc3Q0MiJHMEUCIQD0sD5xb1/98Ff6878zhF3lPg+VFThrxNc0RfAXX82kQIgX1411shYmS2TL4Ft"
5   region = "us-east-1"
6 }
```

The terminal window at the bottom shows the command 'terraform plan' being executed. The output indicates that Terraform will create an 'aws_s3_bucket.mohit' resource with the following attributes:

```
# aws_s3_bucket.mohit will be created
+ resource "aws_s3_bucket" "mohit" {
+   acceleration_status = (known after apply)
+   acl                 = "public-read"
+   arn                 = (known after apply)
+   bucket              = "my-bj-terraform-test-bucket"
+   bucket_domain_name = (known after apply)
+   bucket_prefix       = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy       = false
+   hosted_zone_id      = (known after apply)
+   id                  = (known after apply)
+   object_lock_enabled = (known after apply)
+   policy              = (known after apply)
+   region              = (known after apply)
}
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left displaying the file structure under 'TERRAFORM SCRIPTS'. The file 's3.tf' is selected and open in the editor. The code in 's3.tf' defines an AWS S3 bucket resource with the following details:

```
1 resource "aws_s3_bucket" "mohit" {
2   bucket = "mohit-54"
3
4   tags = {
5     Name       = "My bucket"
6     Environment = "Dev"
7   }
8 }
9
```

The terminal window at the bottom shows the command 'terraform apply' being executed. The output indicates that Terraform will create an 'aws_s3_bucket.mohit' resource with the following attributes:

```
# aws_s3_bucket.mohit will be created
+ resource "aws_s3_bucket" "mohit" {
+   acceleration_status = (known after apply)
+   acl                 = (known after apply)
+   arn                 = (known after apply)
+   bucket              = "mohit-54"
+   bucket_domain_name = (known after apply)
+   bucket_prefix       = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy       = false
+   hosted_zone_id      = (known after apply)
+   id                  = (known after apply)
+   object_lock_enabled = (known after apply)
+   policy              = (known after apply)
+   region              = (known after apply)
}
```

Browser tabs: DomDine - Google, Launch AWS Academy, S3 buckets | S3 | us-east-1, Untitled document, New Tab, Install | Terraform | H, ...

Address bar: us-east-1.console.aws.amazon.com/s3/home?region=us-east-1#

Services: Search [Alt+S]

Region: N. Virginia | User: voclabs/user3402721-2022.mohit.shadija@ves.ac.in @ 2292-9730-...

Amazon S3

- Buckets
- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3
- Block Public Access settings for this account
- Storage Lens
 - Dashboards
 - Storage Lens groups
 - AWS Organizations settings
- Feature spotlight

Account snapshot - updated every 24 hours All AWS Regions [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets | Directory buckets

General purpose buckets (2) All AWS Regions [Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	mohit-54	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 10, 2024, 11:25:17 (UTC+05:30)
<input type="radio"/>	mohitshadija05	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 1, 2024, 15:08:22 (UTC+05:30)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
}
}

- versioning {
  - enabled = false -> null
  - mfa_delete = false -> null
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_s3_bucket.mohit: Destroying... [id=mohit-54]
aws_s3_bucket.mohit: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
PS C:\terraform scripts>
```