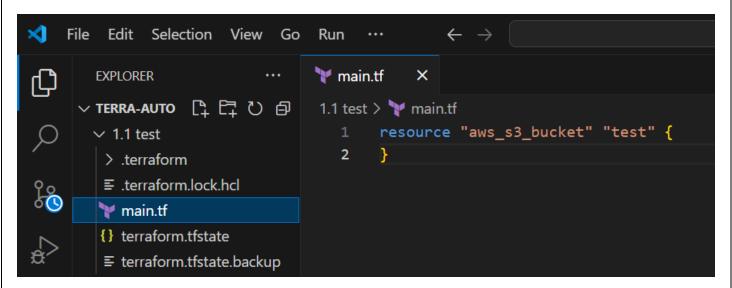
Terra-Auto CMD

Task 1.1 → Using terraform import

I created an empty resource block for importing an existing S3 bucket.



→ aws configure

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR AZURE

• PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto> cd '.\1.1 test\'
• PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.1 test> aws configure
AWS Access Key ID [****************************
AWS Secret Access Key [*****************************
Default region name [us-east-1]:
Default output format [json]:
```

→ terraform init

PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.1 test> terraform initInitializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.69.0...
- Installed hashicorp/aws v5.69.0 (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.

→ terraform import aws_s3_bucket.test bucket-for-terra-auto

```
• PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.1 test> terraform import aws_s3_bucket.test bucket-for-terra-auto aws_s3_bucket.test: Importing from ID "bucket-for-terra-auto"... aws_s3_bucket.test: Import prepared!
Prepared aws_s3_bucket for import
aws_s3_bucket.test: Refreshing state... [id=bucket-for-terra-auto]

Import successful!

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
```

→ terraform plan

```
PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.1 test> terraform plan
  aws_s3_bucket.test: Refreshing state... [id=bucket-for-terra-auto]
 Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
 symbols:
    ∼ update in-place
 Terraform will perform the following actions:
   # aws_s3_bucket.test will be updated in-place
    ~ resource "aws_s3_bucket" "test" {
                                     = false
       + force_destroy
                                    = "bucket-for-terra-auto"
                                     = {
       ~ tags
           - "terra-auto" = "testing" -> null
       ~ tags_all
             "terra-auto" = "testing" -> null
                                                                                                     Activate Windows
```

→ terraform apply --auto-approve

```
PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.1 test> terraform apply --auto-approve
 aws_s3_bucket.test: Refreshing state... [id=bucket-for-terra-auto]
 Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
 symbols:
   update in-place
 Terraform will perform the following actions:
   # aws_s3_bucket.test will be updated in-place
   ~ resource "aws_s3_bucket" "test" {
       + force_destroy
                                    = false
                                    = "bucket-for-terra-auto"
         id
       tags
            "terra-auto" = "testing" -> null
                                                                                                    Activate Windows
       ~ tags_all
             "terra-auto" = "testing" -> null
                                                                                                    Go to Settings to activate Window
```