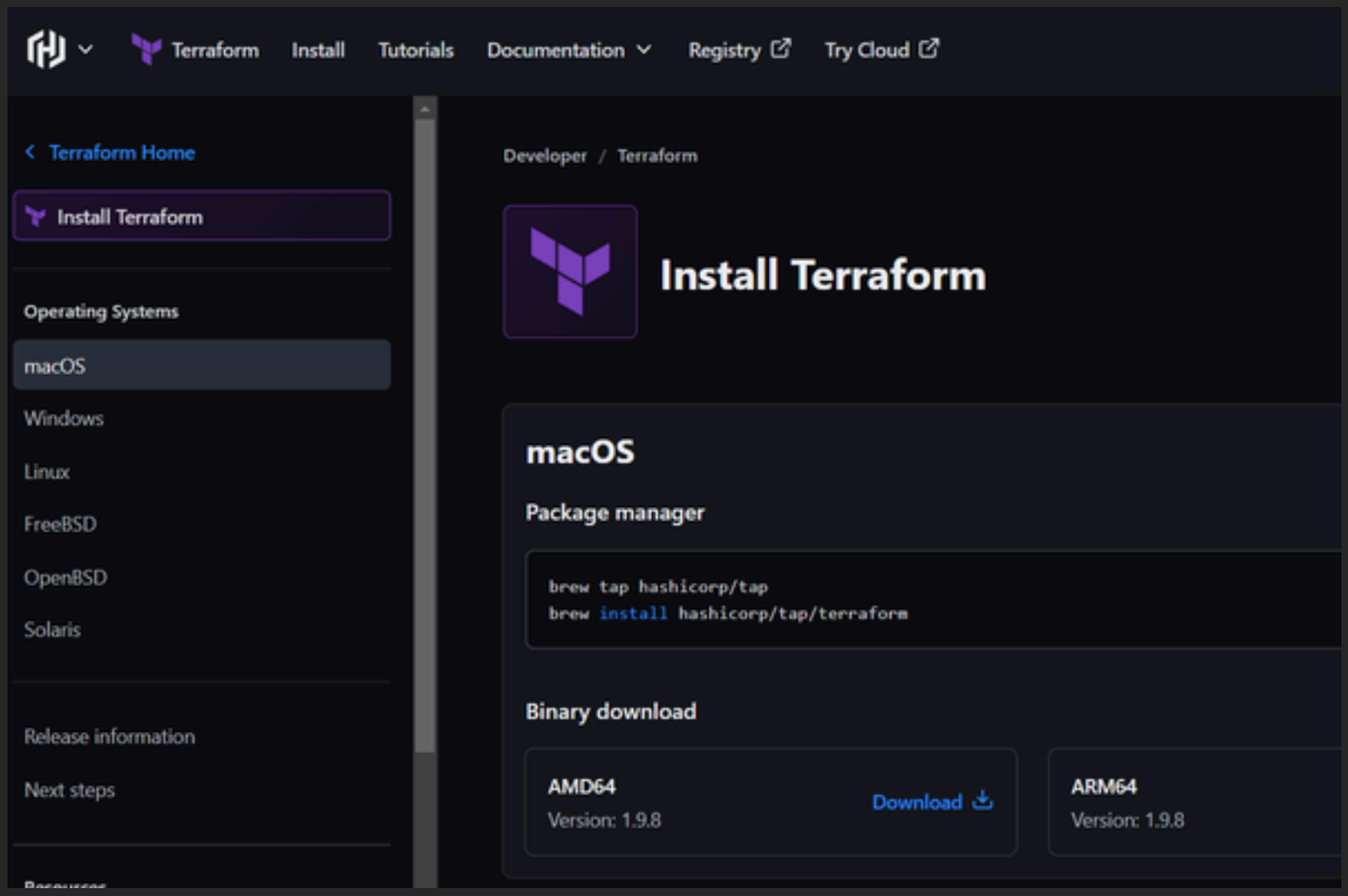
Automated Deployment of EC2 and S3 Infrastructure Using Terraform

Install Terraform:



Configure AWS provider

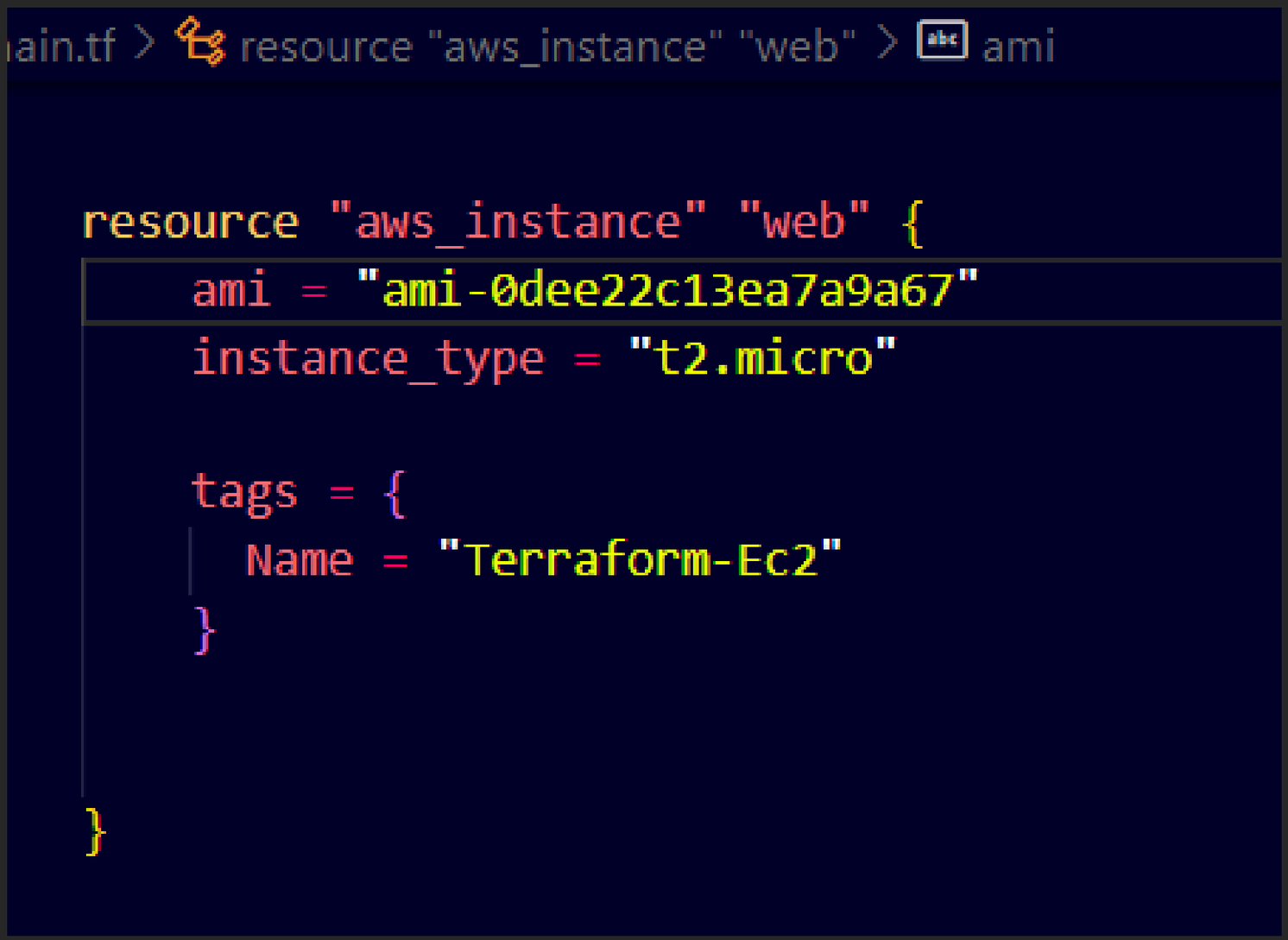
provider.tf > ...

```hcl
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "ap-south-1"
  access_key = ""
  secret_key = ""

}
```

EC2 Configuration File



S3 Configuration File

```
resource "aws_s3_bucket" "terraform-bucket" {
    bucket = "terraform-bucket-kkrish"


}

resource "aws_s3_object" "bucket-data" {
    bucket = aws_s3_bucket.terraform-bucket.bucket
    source = "./demo-file.txt"
    key = "terraform-data.txt"
}
```

Initialize Terraform

```
kkrish@LAPTOP-LAKNHJC5 MINGW64 ~/Vsc/Terraform-1/aws-s3
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.54.1"...
- Installing hashicorp/aws v5.54.1...
- Installed hashicorp/aws v5.54.1 (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,
```

Plan the Configuration

```
kkrish@LAPTOP-LAKNHJC5 MINGW64 ~/Vsc/Terraform-1/aws-s3
$ terraform plan

Terraform used the selected providers to generate the following execution plan.
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.terraform-bucket will be created
  + resource "aws_s3_bucket" "terraform-bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
```

Apply the Configuration

```
kkrish@LAPTOP-LAKNHJC5 MINGW64 ~/Vsc/Terraform-1/aws-s3
$ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource acti
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.terraform-bucket will be created
  + resource "aws_s3_bucket" "terraform-bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "terraform-bucket-kkrish"
      + 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)
```

Terraform Deployment Results