Create a GitHub Repository Using Terraform

■ Purpose

This runbook explains how to create a GitHub repository using Terraform in a secure, repeatable, and professional way. Credentials are not stored in Terraform code.

■ Why Terraform

Terraform enables GitHub resources to be managed as code, ensuring consistency, traceability, and repeatability across environments.

■ Project Structure

```
main.tf
```

■ Terraform Configuration

```
terraform {
  required_providers {
    github = {
      source  = "integrations/github"
      version = "~> 6.0"
    }
  }
}

provider "github" {}

resource "github_repository" "repo" {
  name       = "my-test-repo"
  visibility = "public"
  auto_init = true
}
```

■ Authentication Method

Authentication is handled via environment variables. This avoids hardcoding secrets and follows industry best practices.

Bash (Linux / macOS):

```
export GITHUB_OWNER="your-github-username"
export GITHUB_TOKEN="YOUR_GITHUB_TOKEN"
```

PowerShell (Windows):

```
$env:GITHUB_OWNER="your-github-username"
$env:GITHUB_TOKEN="YOUR_GITHUB_TOKEN"
```

■ Running Terraform

```
terraform init
terraform plan
terraform apply
```

■ Result

A new public GitHub repository is created with a README file, default branch, and initial commit. Terraform tracks the repository state locally.

■ Summary

This approach demonstrates secure authentication, cross-platform compatibility, and real-world Terraform usage suitable for professional environments.