Exp : Creating Github Repo using Terraform

NOTE:

GitHub and Terraform Experiment: Pre-Experiment Checklist

GitHub Token:** Ensure you have a GitHub personal access token with appropriate permissions (`repo` scope) for accessing repositories.

Terraform Installation:** Confirm that Terraform is installed on your machine and is accessible via the command line.

Repository Naming:** Make sure the repository name you plan to create with Terraform does not already exist in your GitHub account or organization.

Environment Variables:** Set up any required environment variables, such as `GITHUB_TOKEN`, to authenticate with GitHub via Terraform.

Error Handling:** Be prepared to handle and troubleshoot any 404 errors or permission issues related to GitHub API calls.

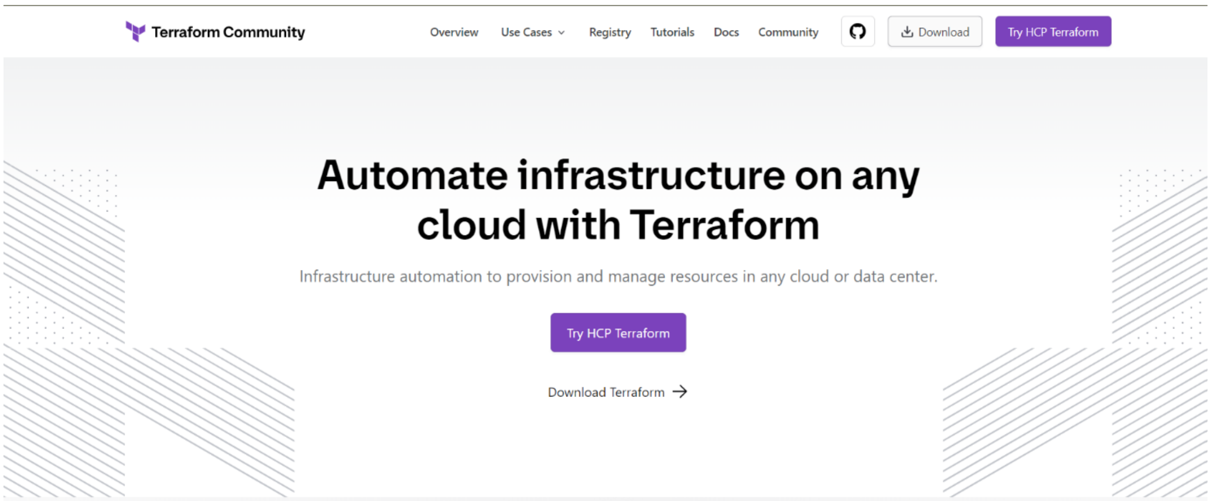
Editor Configuration: It's helpful to have GitHub configured with your editor (e.g., VS Code) for easier management of commits, pushes, and pulls directly from the editor.

Refrences:

https://developer.hashicorp.com/terraform/install?product_intent=terraform#windows

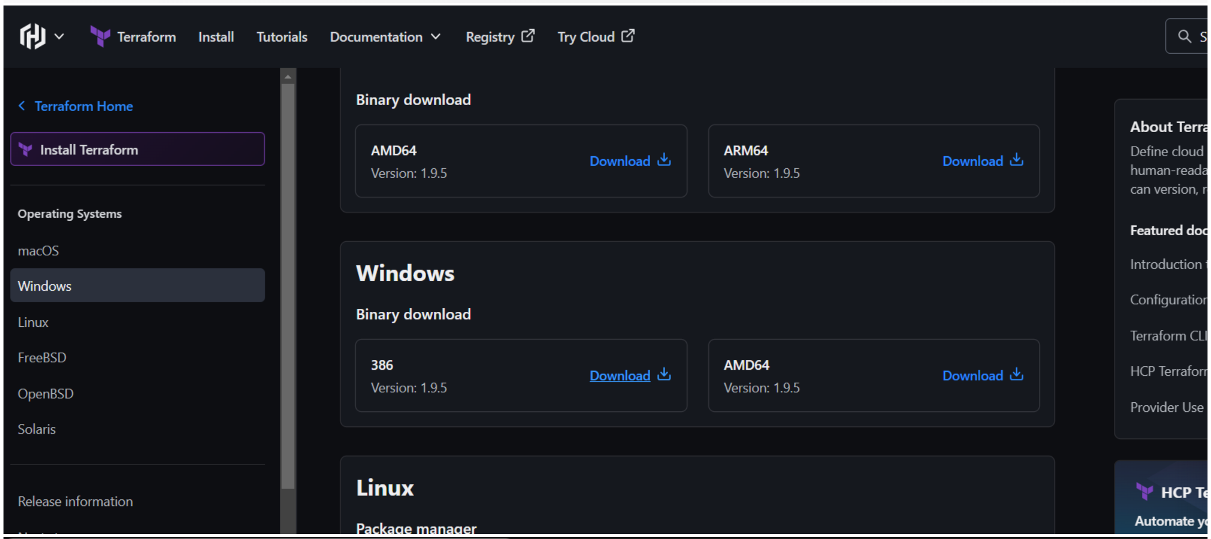
https://registry.terraform.io/providers/integrations/github/latest/docs

https://youtu.be/aHve0Ji13IY?si=ScHDfWw6dqfuoBxz



Automate infrastructure on any cloud with Terraform

Infrastructure automation to provision and manage resources in any cloud or data center.



Binary download

AMD64 Version: 1.9.5 Download	ARM64 Version: 1.9.5 Download

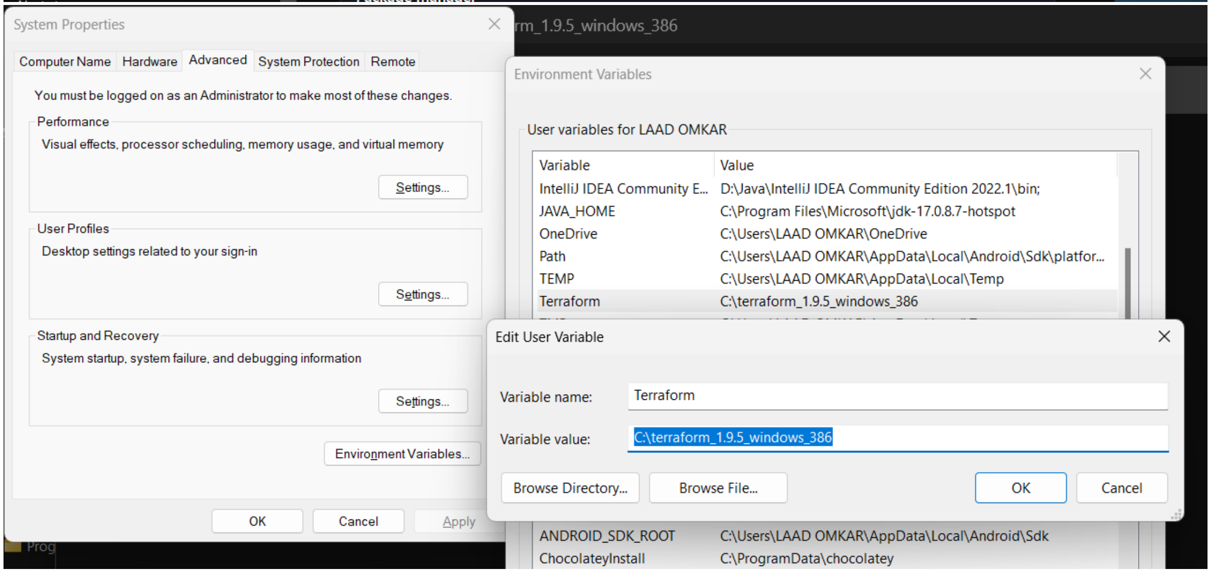
Windows

Binary download

386 Version: 1.9.5 Download	AMD64 Version: 1.9.5 Download

Linux

Package manager



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

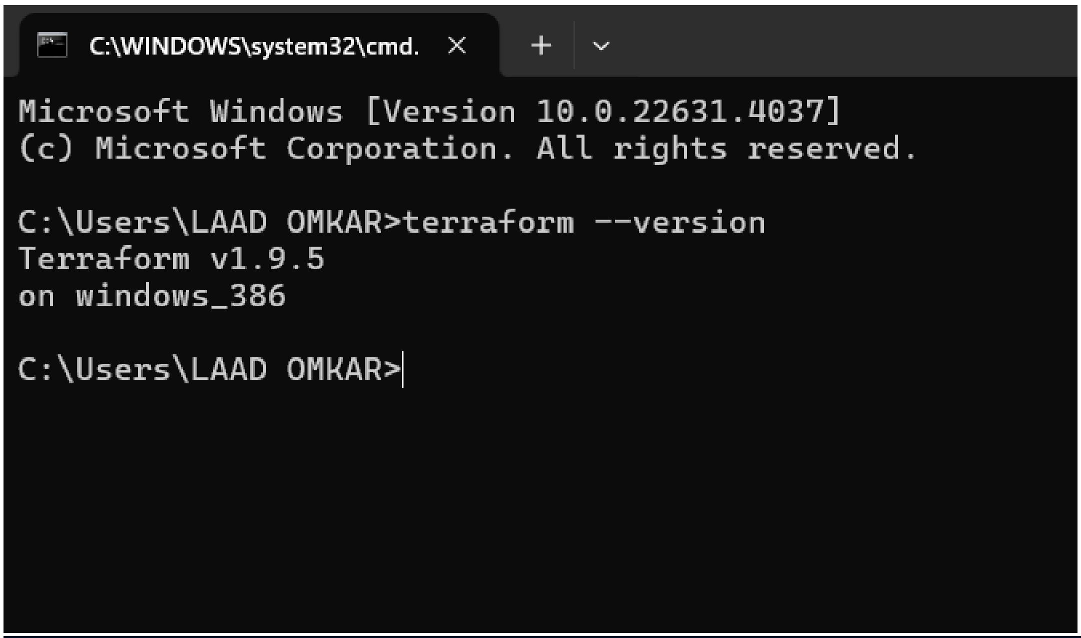
User variables for LAAD OMKAR

Variable	Value
IntelliJ IDEA Community E...	D:\Java\IntelliJ IDEA Community Edition 2022.1\bin;
JAVA_HOME	C:\Program Files\Microsoft\jdk-17.0.8.7-hotspot
OneDrive	C:\Users\LAAD OMKAR\OneDrive
Path	C:\Users\LAAD OMKAR\AppData\Local\Android\Sdk\platfor...
TEMP	C:\Users\LAAD OMKAR\AppData\Local\Temp
Terraform	C:\terraform_1.9.5_windows_386
ANDROID_SDK_ROOT	C:\Users\LAAD OMKAR\AppData\Local\Android\Sdk
ChocolateyInstall	C:\ProgramData\chocolatey

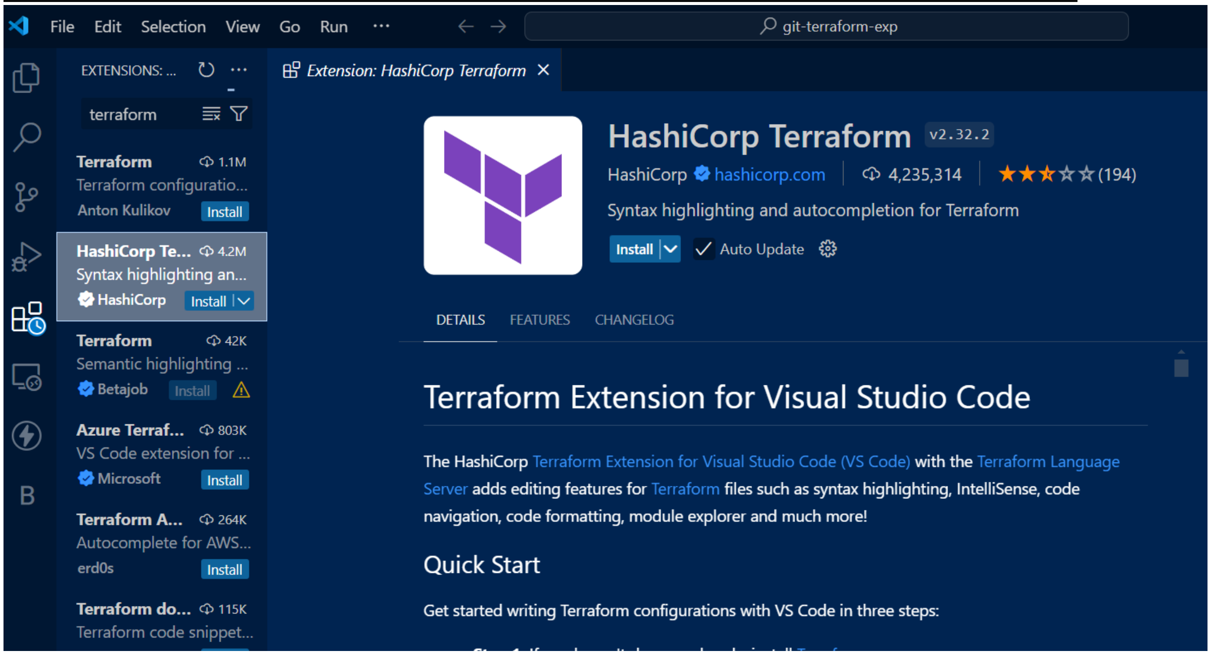
Edit User Variable

Variable name: Terraform

Variable value: C:\terraform_1.9.5_windows_386



```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LAAD OMKAR>terraform --version
Terraform v1.9.5
on windows_386

C:\Users\LAAD OMKAR>
```



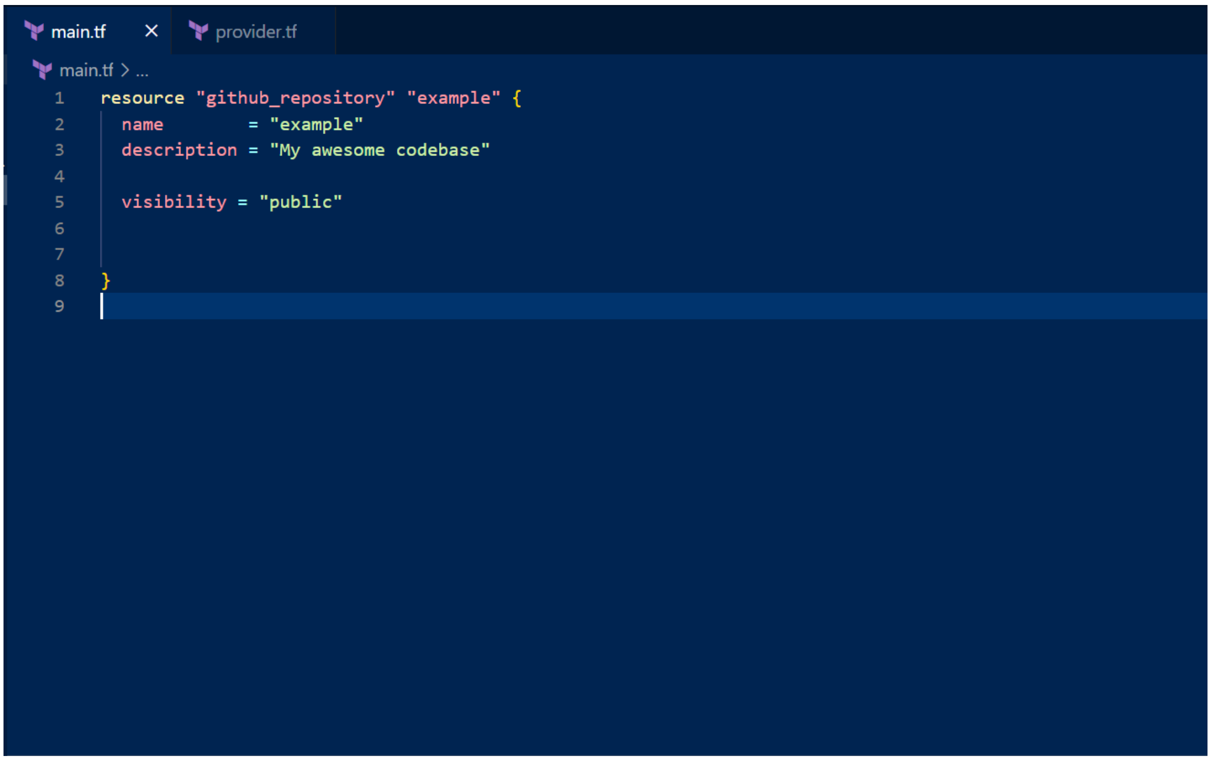
File Edit Selection View Go Run ···
git-terraform-exp
EXTENSIONS: ...
terraform
Terraform 1.1M Terraform configuratio... Anton Kulikov Install
HashiCorp Te... 4.2M Syntax highlighting an... HashiCorp Install
Terraform 42K Semantic highlighting ... Betajob Install
Azure Terraf... 803K VS Code extension for ... Microsoft Install
Terraform A... 264K Autocomplete for AWS... erd0s Install
Terraform do... 115K Terraform code snippet...
Extension: HashiCorp Terraform
HashiCorp Terraform v2.32.2
HashiCorp hashicorp.com 4,235,314 (194)
Syntax highlighting and autocompletion for Terraform
Install Auto Update
DETAILS FEATURES CHANGELOG

Terraform Extension for Visual Studio Code

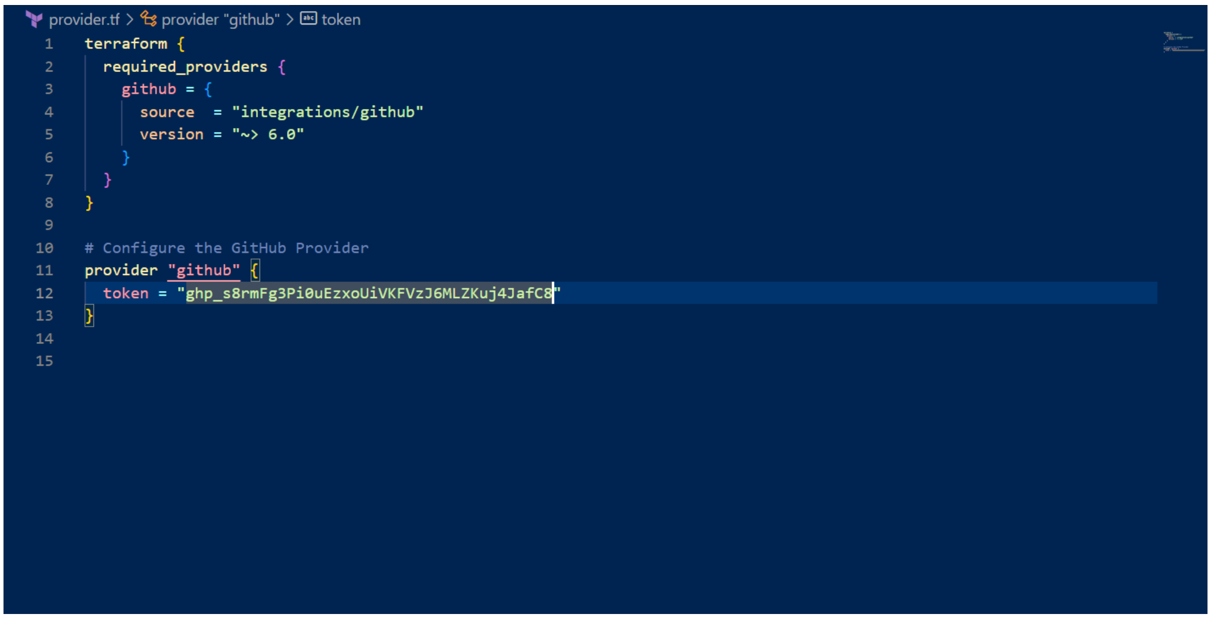
The HashiCorp Terraform Extension for Visual Studio Code (VS Code) with the Terraform Language Server adds editing features for Terraform files such as syntax highlighting, IntelliSense, code navigation, code formatting, module explorer and much more!

Quick Start

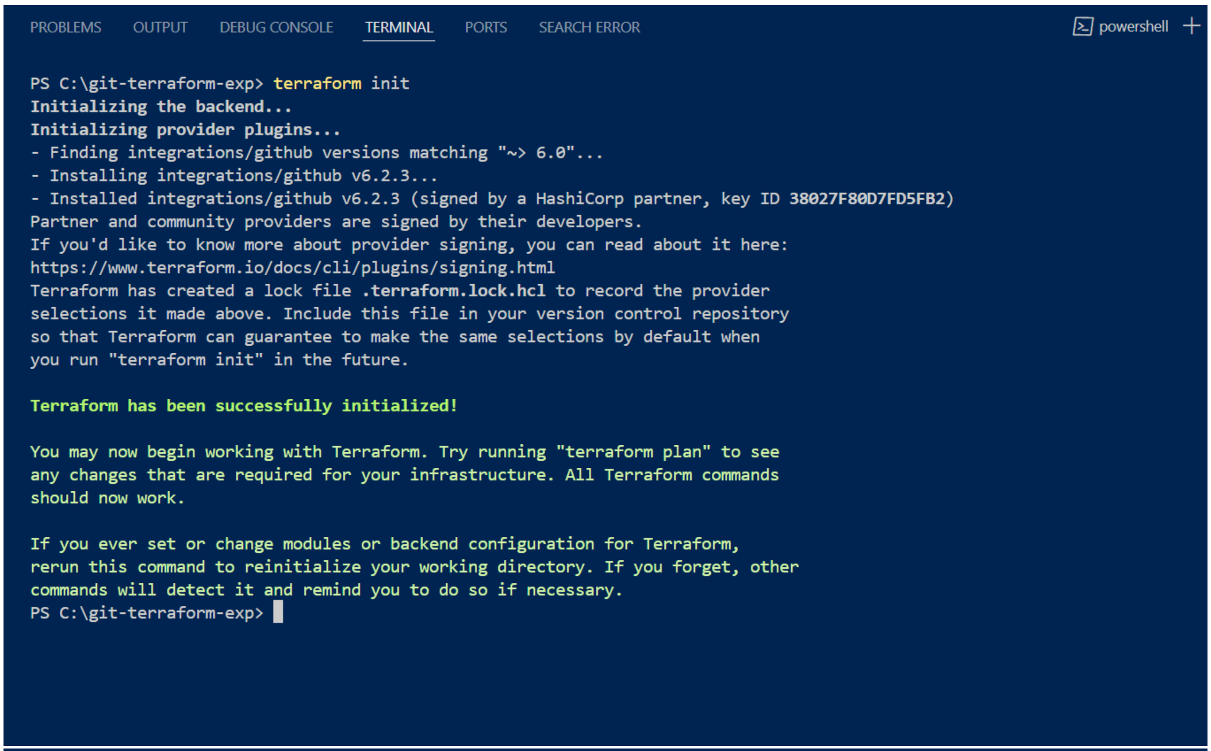
Get started writing Terraform configurations with VS Code in three steps:



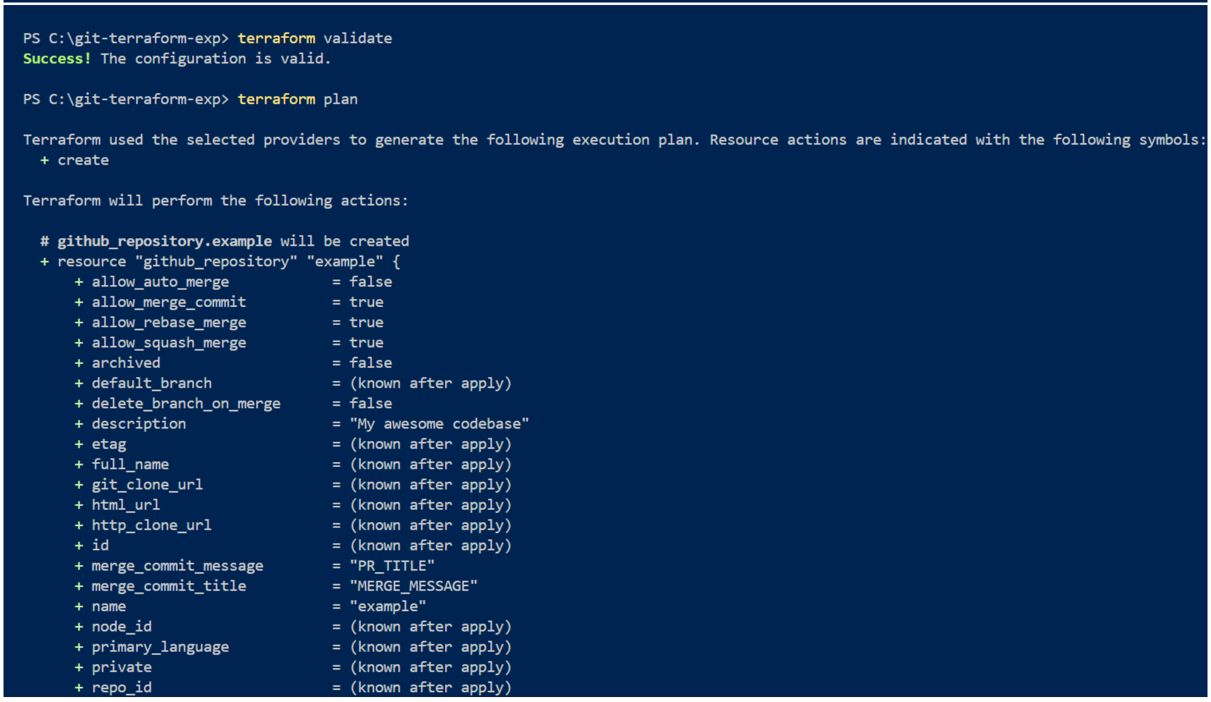
```
resource "github_repository" "example" {
  name        = "example"
  description = "My awesome codebase"

  visibility = "public"


}

```



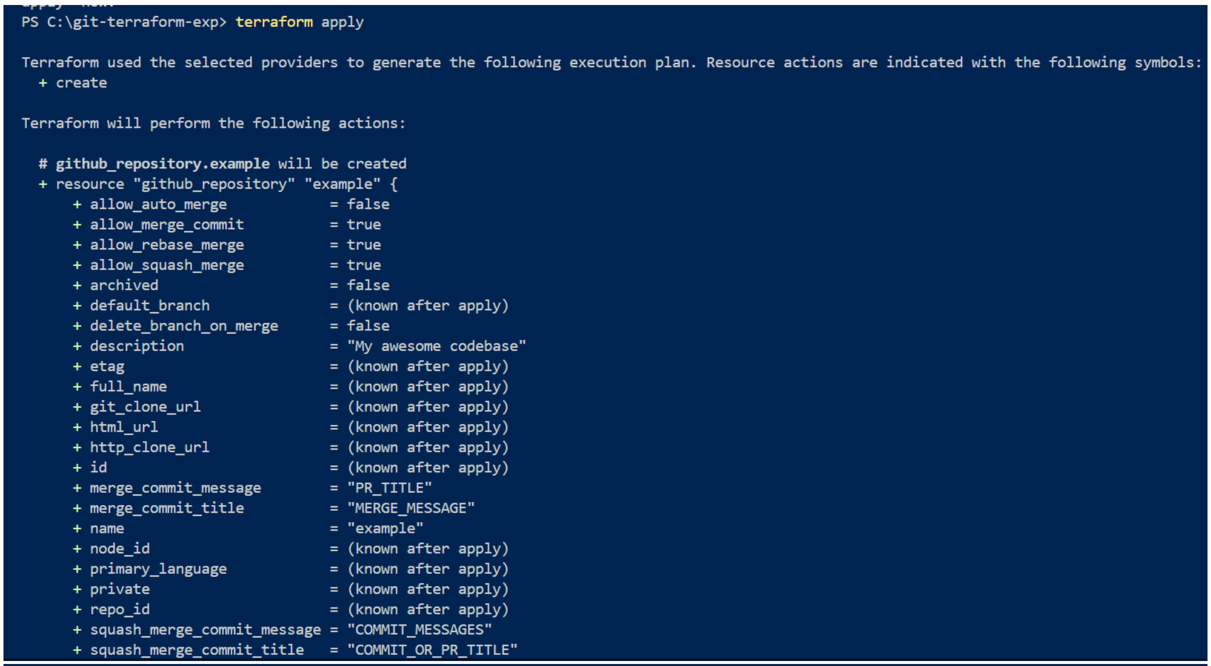
```
terraform {
  required_providers {
    github = {
      source  = "integrations/github"
      version = "~> 6.0"
    }
  }
}

# Configure the GitHub Provider
provider "github" {
  token = "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"
}


```



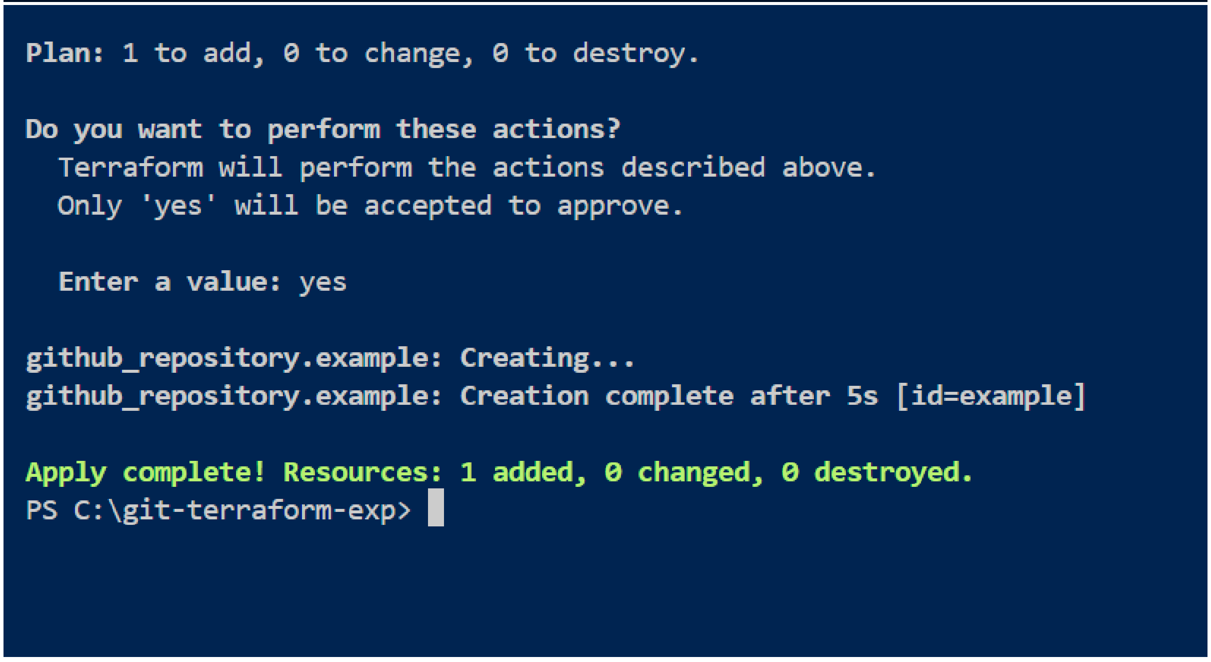
```
PS C:\git-terraform-exp> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding integrations/github versions matching "~> 6.0"...
- Installing integrations/github v6.2.3...
- Installed integrations/github v6.2.3 (signed by a HashiCorp partner, key ID 38027F80D7FD5FB2)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
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.
PS C:\git-terraform-exp>
```



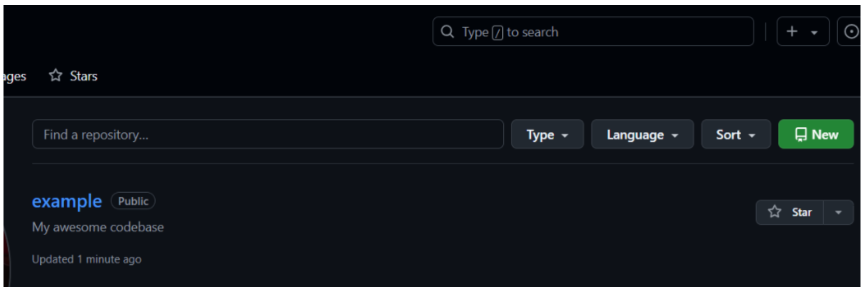
```
PS C:\git-terraform-exp> terraform validate
Success! The configuration is valid.

PS C:\git-terraform-exp> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # github_repository.example will be created
  + resource "github_repository" "example" {
      + allow_auto_merge            = false
      + allow_merge_commit          = true
      + allow_rebase_merge          = true
      + allow_squash_merge          = true
      + archived                    = false
      + default_branch              = (known after apply)
      + delete_branch_on_merge      = false
      + description                 = "My awesome codebase"
      + etag                        = (known after apply)
      + full_name                   = (known after apply)
      + git_clone_url               = (known after apply)
      + html_url                    = (known after apply)
      + http_clone_url              = (known after apply)
      + id                          = (known after apply)
      + merge_commit_message        = "PR_TITLE"
      + merge_commit_title          = "MERGE_MESSAGE"
      + name                        = "example"
      + node_id                     = (known after apply)
      + primary_language            = (known after apply)
      + private                     = (known after apply)
      + repo_id                     = (known after apply)
```



```
PS C:\git-terraform-exp> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # github_repository.example will be created
  + resource "github_repository" "example" {
      + allow_auto_merge            = false
      + allow_merge_commit          = true
      + allow_rebase_merge          = true
      + allow_squash_merge          = true
      + archived                    = false
      + default_branch              = (known after apply)
      + delete_branch_on_merge      = false
      + description                 = "My awesome codebase"
      + etag                        = (known after apply)
      + full_name                   = (known after apply)
      + git_clone_url               = (known after apply)
      + html_url                    = (known after apply)
      + http_clone_url              = (known after apply)
      + id                          = (known after apply)
      + merge_commit_message        = "PR_TITLE"
      + merge_commit_title          = "MERGE_MESSAGE"
      + name                        = "example"
      + node_id                     = (known after apply)
      + primary_language            = (known after apply)
      + private                     = (known after apply)
      + repo_id                     = (known after apply)
      + squash_merge_commit_message = "COMMIT_MESSAGES"
      + squash_merge_commit_title   = "COMMIT_OR_PR_TITLE"
```



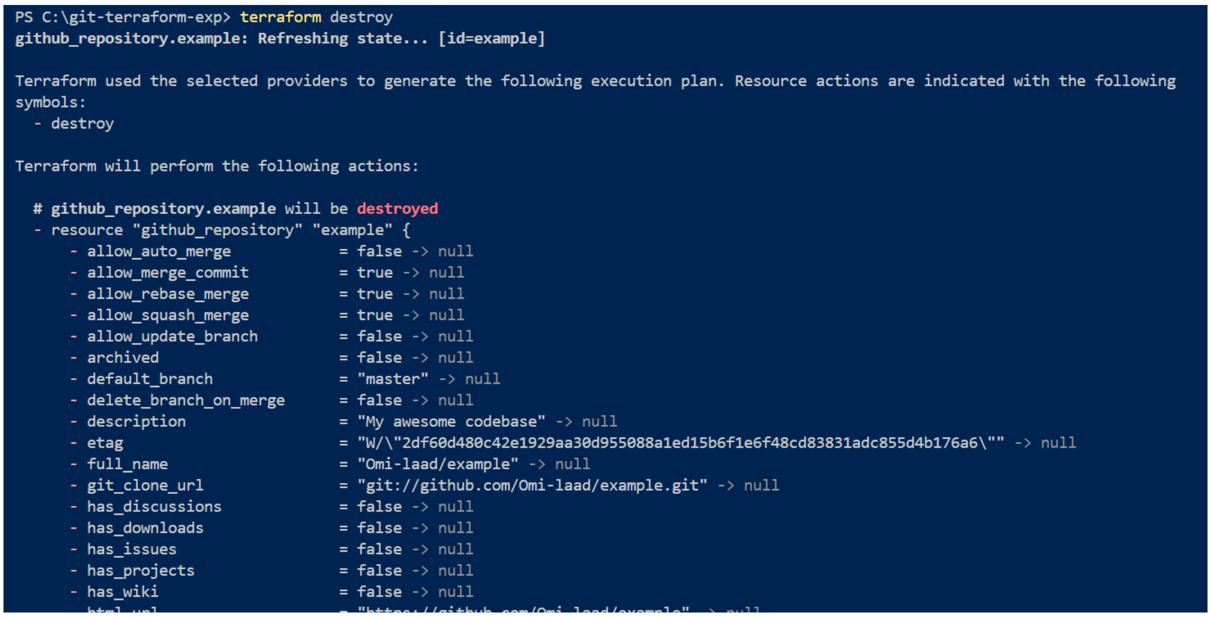
```
Plan: 1 to add, 0 to change, 0 to destroy.

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

github_repository.example: Creating...
github_repository.example: Creation complete after 5s [id=example]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\git-terraform-exp>
```



example Public

My awesome codebase

Updated 1 minute ago



```
PS C:\git-terraform-exp> terraform destroy
github_repository.example: Refreshing state... [id=example]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  - destroy

Terraform will perform the following actions:

  # github_repository.example will be destroyed
  - resource "github_repository" "example" {
      - allow_auto_merge            = false -> null
      - allow_merge_commit          = true -> null
      - allow_rebase_merge          = true -> null
      - allow_squash_merge          = true -> null
      - allow_update_branch         = false -> null
      - archived                    = false -> null
      - default_branch              = "master" -> null
      - delete_branch_on_merge      = false -> null
      - description                 = "My awesome codebase" -> null
      - etag                        = "W/\"2df60d480c42e1929aa30d955088a1ed15b6f1e6f48cd83831adc855d4b176a6\"" -> null
      - full_name                   = "Omi-laad/example" -> null
      - git_clone_url               = "git://github.com/Omi-laad/example.git" -> null
      - has_discussions             = false -> null
      - has_downloads               = false -> null
      - has_issues                  = false -> null
      - has_projects                = false -> null
      - has_wiki                    = false -> null
      - html_url                    = "https://github.com/Omi-laad/example" -> null
```