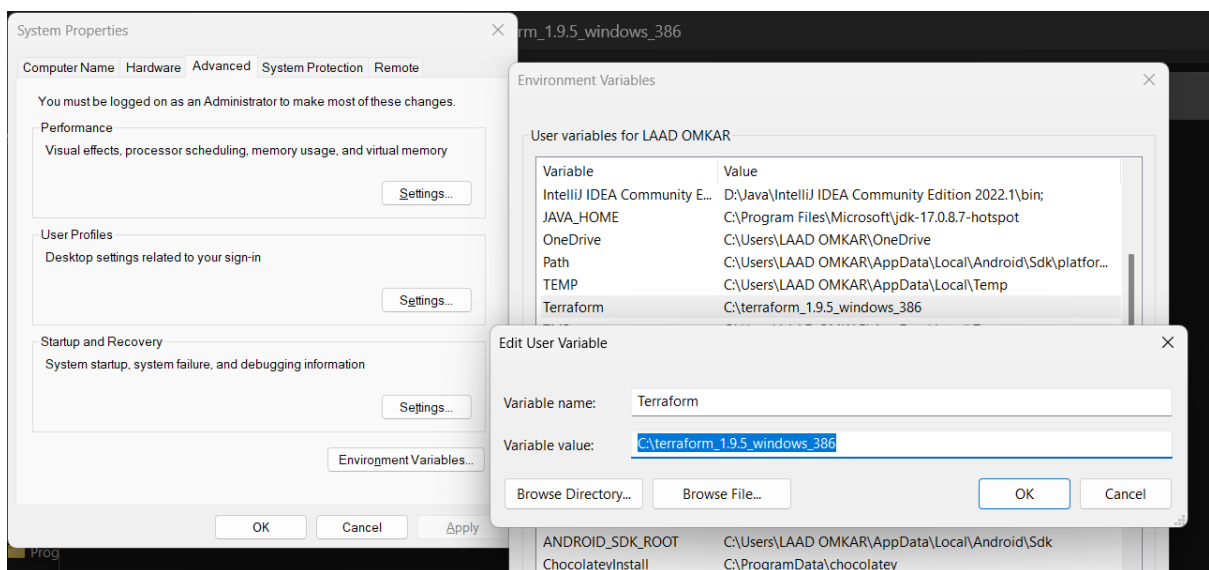
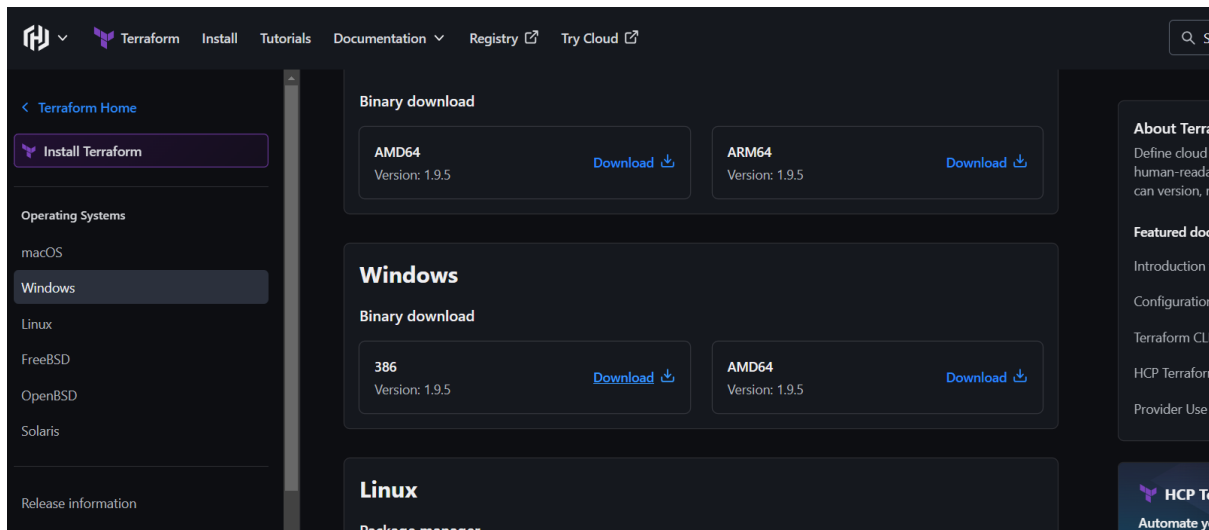
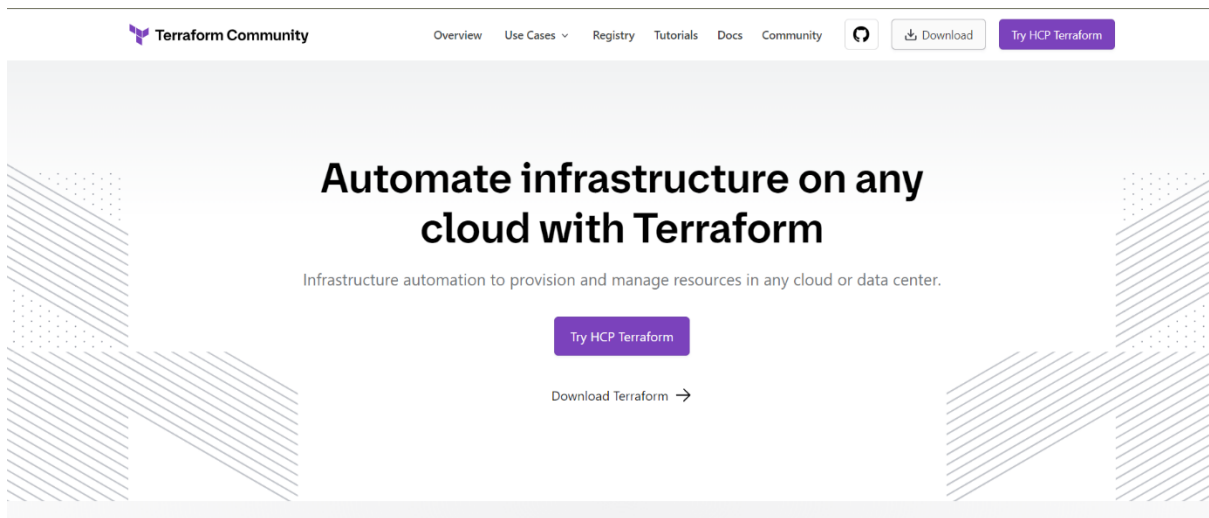


## Exp : Creating Github Repo using Terraform

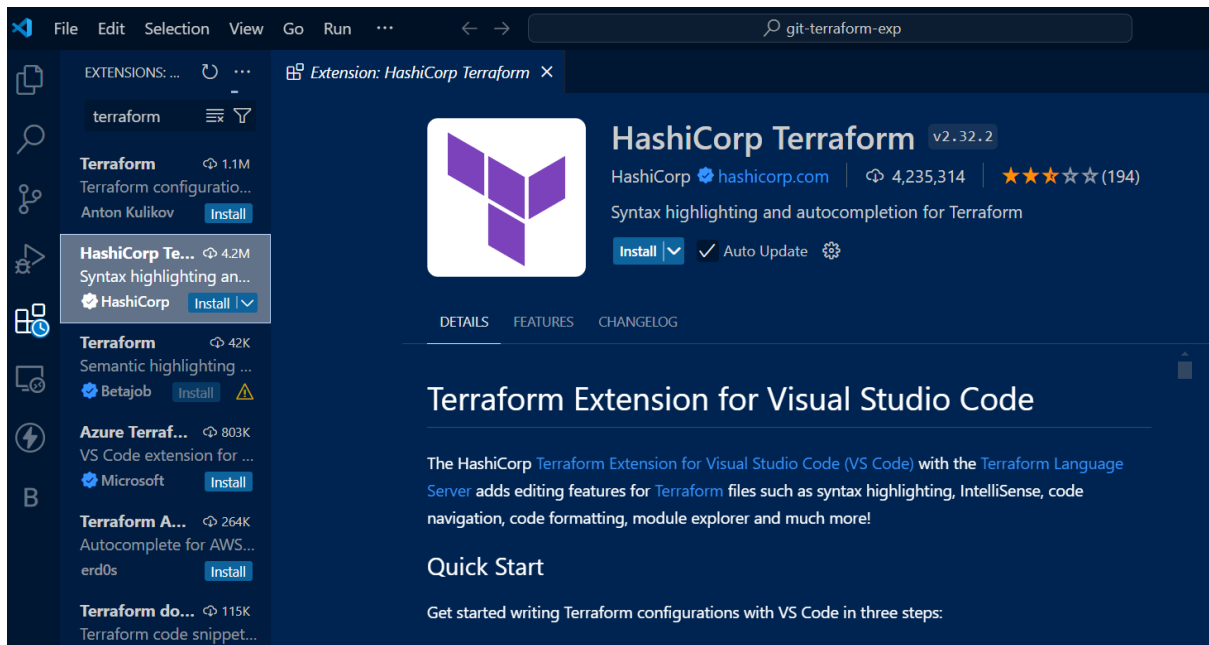


```
C:\WINDOWS\system32\cmd. x + v

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>
```



```
main.tf  X  provider.tf

main.tf > ...
1 resource "github_repository" "example" {
2     name      = "example"
3     description = "My awesome codebase"
4
5     visibility = "public"
6
7
8 }
9
```

```
provider.tf > provider "github" > token

1 terraform {
2     required_providers {
3         github = {
4             source = "integrations/github"
5             version = "~> 6.0"
6         }
7     }
8 }
9
10 # Configure the GitHub Provider
11 provider "github" {
12     token = "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"
13 }
14
15
```

```
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.

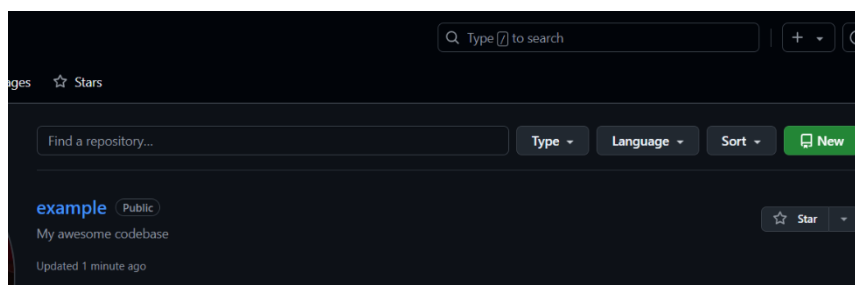
Enter a value: yes

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>



```
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
```