

Name: Manav Jawrani

Roll No.: 19

Subject: Advanced DevOps

Experiment No.: 5

Experiment 5

Aim: To understand terraform lifecycle, core concepts / terminologies and install it on a linux machine / windows machine

Theory:

- What is terraform?

Terraform is IaC tool, used primarily by Dev Ops teams to automate various infrastructure tasks. The provisioning of cloud resources, for instance is one of the main use cases for Terraform. Terraform allows you to describe your complete infrastructure in the form of code. Even if your servers come from different providers such as AWS or Azure. Terraform helps you build and manage these resources in parallel across providers.

- What are the features of terraform?

1. Infrastructure as Code (IaC)
2. Execution Plans
3. Resource Graph
4. Change Automation

• Testaform Core Elements and Providers

Testaform consists of two chief components:

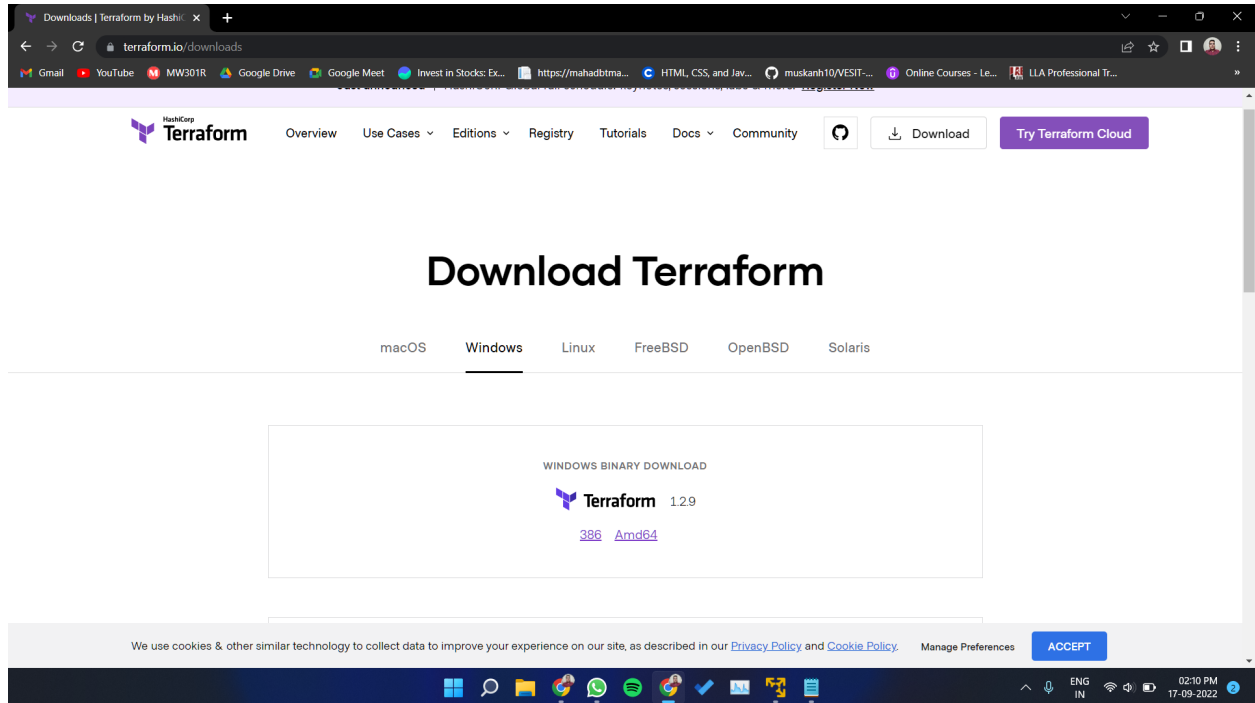
1. Testaform core:- This component monitors the loading and interpolation of configurations files, resource plan executions, state management features and resource graphs. Testaform core is made up of compiled binaries written in Go language.
2. Testaform plugin:- The plugin defines resources for specific services, including initializing the libraries used to make API calls and authentication infrastructure providers. As is the case with Testaform core, Testaform plugins are written in Go programming language as executable binaries for either a specific server.

Implementation: A) Installation and Configuration of Terraform in Windows

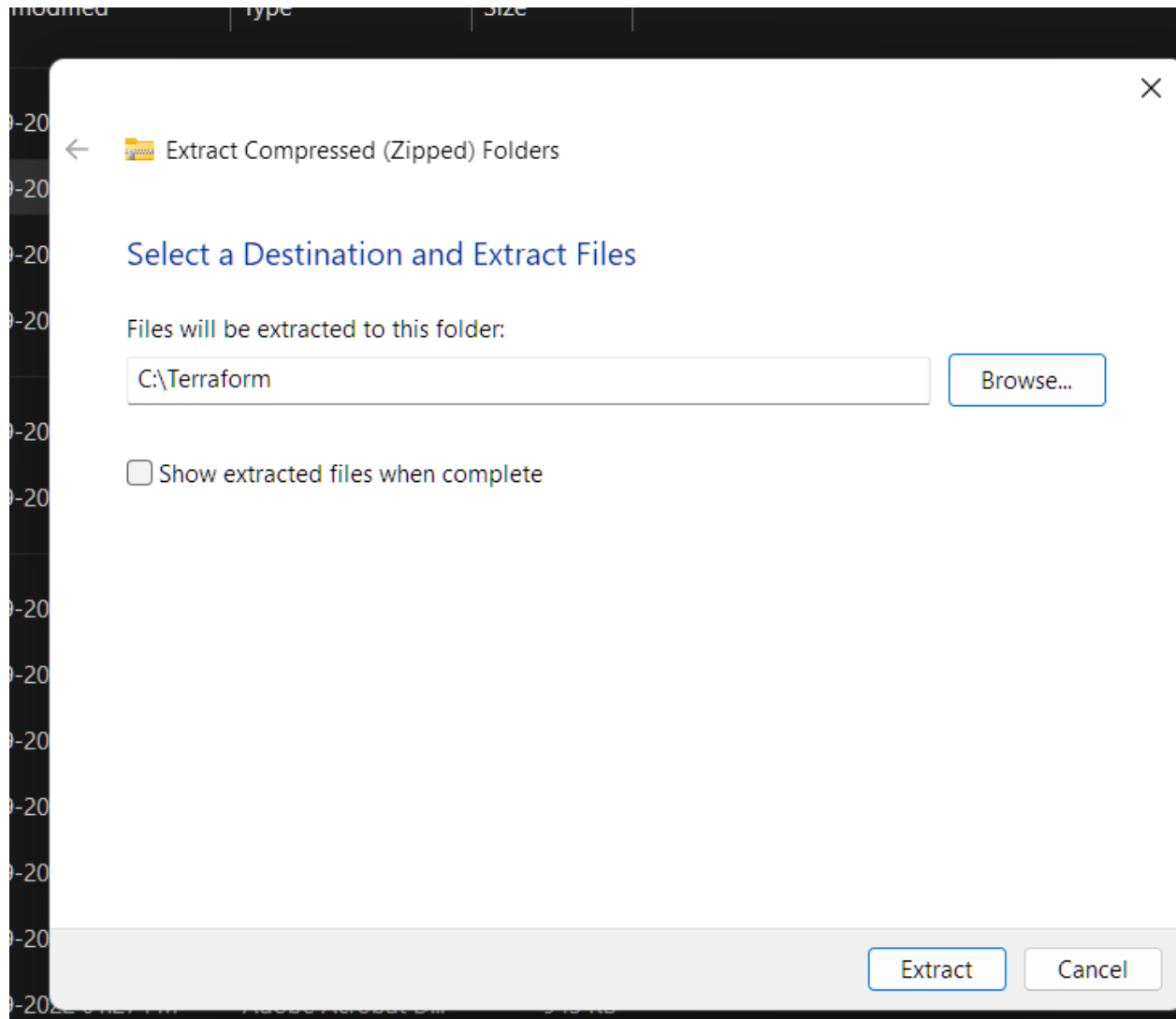
Step 1: Download terraform

To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website [Downloads | Terraform by HashiCorp](https://www.terraform.io/downloads)

Select the Operating System Windows and download the file.



Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



[illegible]

Step 4: Open Powershell in admin and type “terraform” to check the installation

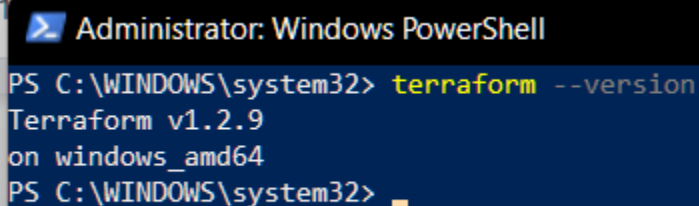
```
PS C:\WINDOWS\system32> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init       Prepare your working directory for other commands
  validate   Check whether the configuration is valid
  plan       Show changes required by the current configuration
  apply      Create or update infrastructure
  destroy    Destroy previously-created infrastructure

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  output     Show output values from your root module
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint      Mark a resource instance as not fully functional
  test       Experimental support for module integration testing
  untaint    Remove the 'tainted' state from a resource instance
  version    Show the current Terraform version
  workspace  Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
               given subcommand.
  -help      Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.
PS C:\WINDOWS\system32>
```



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> terraform --version
Terraform v1.2.9
on windows_amd64
PS C:\WINDOWS\system32>
```

****Note:** If any error comes, then please recheck or set the path of Terraform in Environment variable again.**

Conclusion :

In this experiment, I learned about terraform and what type of tool it is, which is an IaC. I observed some terraform code concepts such as the programming it employs, how it describes infrastructure code and the execution plan. Following that we installed it on our windows machine