Experiment 5

AIM: To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine and Windows.

Theory:

Terraform is an infrastructure as code (IaC) tool developed by HashiCorp that allows users to define and manage cloud and on-premises resources using declarative configuration files. Understanding its lifecycle and core concepts is essential for effectively utilizing Terraform in managing infrastructure.

Terraform Lifecycle

Terraform manages resources through a defined lifecycle that includes three primary stages:

1. Apply (Create): This stage involves creating the resource as defined in the configuration files. Terraform communicates with the cloud provider's API to provision the resource.
2. Update: In this stage, Terraform makes modifications to existing resources. This can involve incremental changes or complete replacements, depending on the nature of the update.
3. Destroy: This final stage removes the resource from the environment, ensuring that it is no longer managed by Terraform.

Terraform also provides a lifecycle meta-argument that allows users to control these stages more granularly. Options within this meta-argument include:

- create_before_destroy: Ensures a new instance is created before the old one is destroyed, which is useful for maintaining service availability during updates.
- prevent_destroy: Prevents accidental deletion of critical resources.
- ignore_changes: Allows Terraform to ignore changes made outside of its management for specified fields or entire objects.

Core Concepts

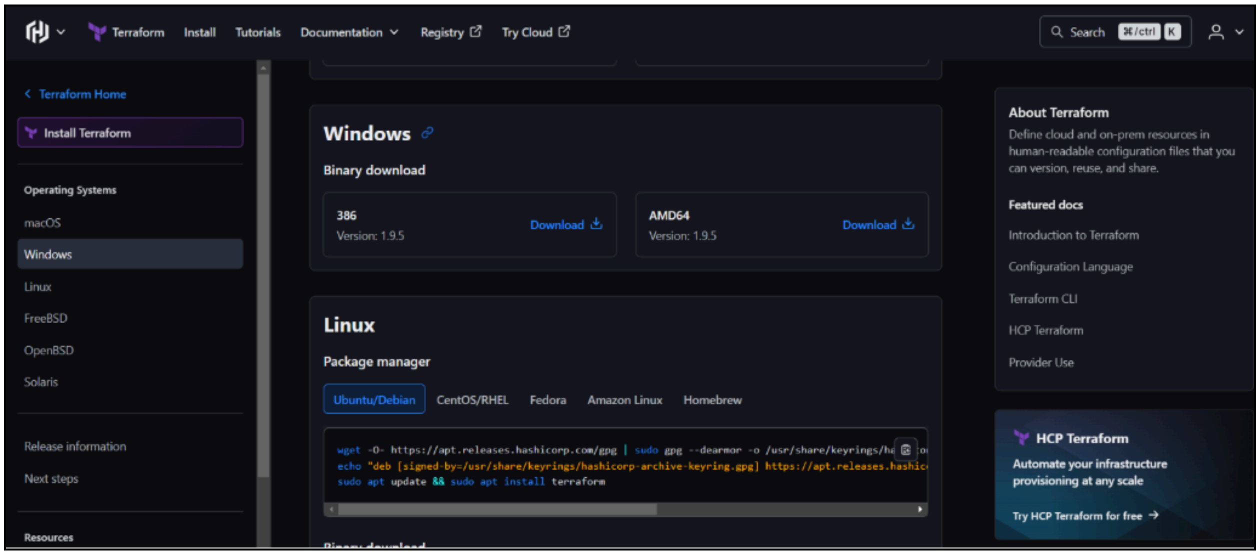
Several key concepts underpin Terraform's functionality:

- Configuration Files: These files, written in HashiCorp Configuration Language (HCL), define the desired state of the infrastructure. They describe resources, their properties, and relationships.
- Resources: The fundamental building blocks in Terraform, representing various infrastructure components like virtual machines, storage accounts, and networking configurations.
- Data Sources: These provide external data to Terraform configurations, allowing users to reference existing resources or configurations from other projects.
- Modules: Modules are reusable packages of Terraform configurations that enable users to group related resources for better organization and reusability. They can be shared across projects and teams, promoting best practices and consistency.
- State Management: Terraform maintains a state file that acts as a source of truth for the infrastructure. This file tracks the current state of resources, enabling Terraform to determine the necessary changes during the apply phase.
- Execution Plan: Before applying changes, Terraform generates an execution plan that outlines what actions will be taken, allowing users to review and approve changes before they are executed.

By mastering these concepts and understanding the lifecycle of resources, users can effectively leverage Terraform to automate and manage their infrastructure in a safe and efficient manner.

Implementation:

Step 1: Go to Terraform website and download 386 .



Windows

Binary download

386 Version: 1.9.5	Download
AMD64 Version: 1.9.5	Download

Linux

Package manager

Ubuntu/Debian CentOS/RHEL Fedora Amazon Linux Homebrew

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/ha
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashic
sudo apt update && sudo apt install terraform
```

About Terraform

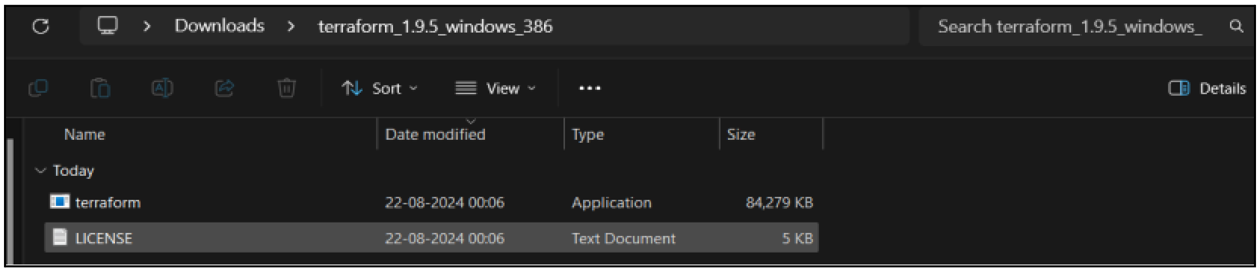
Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Featured docs

Introduction to Terraform
Configuration Language
Terraform CLI
HCP Terraform
Provider Use

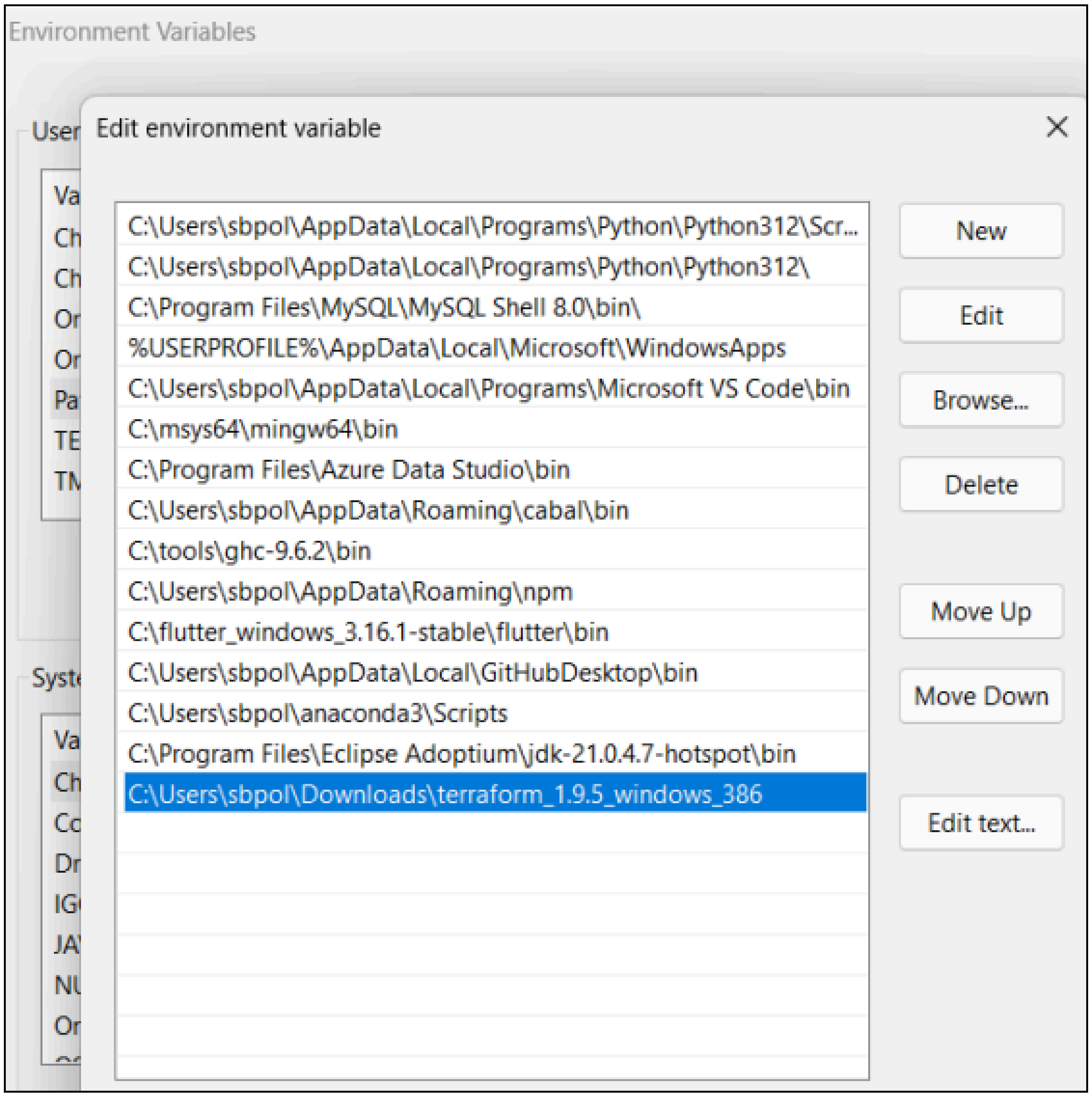
HCP Terraform
Automate your infrastructure provisioning at any scale
Try HCP Terraform for free

Step 2 : Extract the zip file and copy the path of the file.


Downloads > terraform_1.9.5_windows_386
Search terraform_1.9.5_windows_
Sort
View
Details

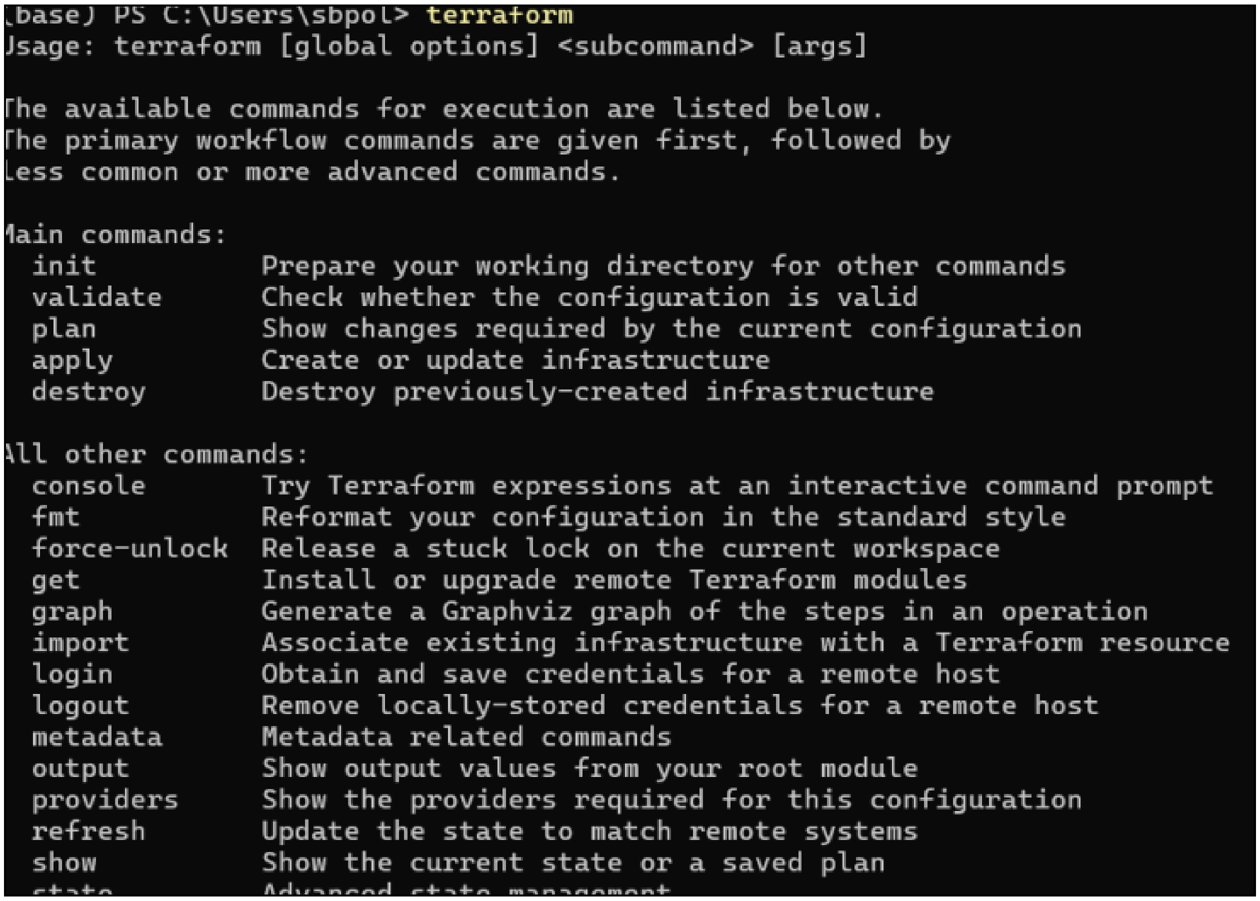
Name	Date modified	Type	Size
Today			
terraform	22-08-2024 00:06	Application	84,279 KB
LICENSE	22-08-2024 00:06	Text Document	5 KB

Step 3: Edit environment variables and paste copied path.



```
C:\Users\sbpol\AppData\Local\Programs\Python\Python312\Scr...
C:\Users\sbpol\AppData\Local\Programs\Python\Python312\
C:\Program Files\MySQL\MySQL Shell 8.0\bin\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\sbpol\AppData\Local\Programs\Microsoft VS Code\bin
C:\msys64\mingw64\bin
C:\Program Files\Azure Data Studio\bin
C:\Users\sbpol\AppData\Roaming\cabal\bin
C:\tools\ghc-9.6.2\bin
C:\Users\sbpol\AppData\Roaming\npm
C:\flutter_windows_3.16.1-stable\flutter\bin
C:\Users\sbpol\AppData\Local\GitHubDesktop\bin
C:\Users\sbpol\anaconda3\Scripts
C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin
C:\Users\sbpol\Downloads\terraform_1.9.5_windows_386
```

Step 4 : Make sure Terraform is installed successfully.

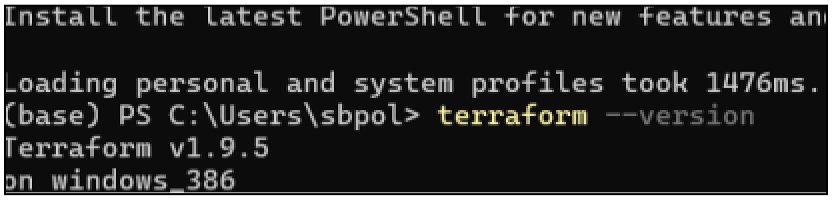


```
(base) PS C:\Users\sbpol> terraform
Usage: terraform [global options] <subcommand> [args]

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
  metadata      Metadata related commands
  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
```



```
Install the latest PowerShell for new features an

Loading personal and system profiles took 1476ms.
(base) PS C:\Users\sbpol> terraform --version
Terraform v1.9.5
on windows_386
```