



Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra

Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Experiment 05

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

Roll No.	53
Name	Aryan Deepak Saraf
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO3: To apply best practices for managing infrastructure as code environments and use terraform to define and deploy cloud infrastructure.
Grade:	

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

THEORY :

Terraform is an open-source Infrastructure as Code (IaC) tool that allows users to define and manage infrastructure using a high-level configuration language called HashiCorp Configuration Language (HCL). By treating infrastructure as code, Terraform enables the automation of provisioning, modification, and management of resources across multiple cloud providers, ensuring consistency and repeatability.

Core Concepts and Terminologies

1. **Infrastructure as Code (IaC):** Terraform allows infrastructure to be defined, versioned, and managed as code, making it easier to automate the provisioning of resources like virtual machines, networks, and storage.
2. **Terraform Configuration:** Configurations are written in HCL or JSON, saved in `.tf` files, and describe the desired state of your infrastructure, including the resources to be created and their properties.
3. **Providers:** Providers are plugins that enable Terraform to interact with various cloud platforms and services. They manage specific resources via APIs.
4. **Resources:** Resources represent infrastructure components such as compute instances and databases. Terraform manages their lifecycle, ensuring they are created, updated, or destroyed based on the configuration.
5. **State:** Terraform's state file records the current state of the infrastructure, ensuring that Terraform accurately tracks resource changes and maintains the desired state.
6. **Execution Plan:** The execution plan (`terraform plan`) outlines the changes Terraform will make to achieve the desired state, allowing users to review the plan before applying it.
7. **Apply:** The `terraform apply` command executes the changes specified in the execution plan, updating the infrastructure to match the desired state.
8. **Modules:** Modules are reusable blocks of configuration that help organize and manage resources, promoting code reuse and consistency.

Terraform Lifecycle

1. **Initialization (`terraform init`):** Initializes the working directory, downloads provider plugins, and prepares the environment for further commands.
2. **Planning (`terraform plan`):** Generates an execution plan by comparing the current state with the desired state defined in the configuration, detailing the actions Terraform will take.

3. Applying (`terraform apply`): Executes the changes specified in the execution plan, creating or modifying resources to match the desired state.

Destroying (`terraform destroy`): Deletes all resources managed by the current configuration, useful for completely tearing down an environment.

Download terraform

The screenshot shows the HashiCorp Terraform installation page for Windows. The page is titled "Download terraform" and features a navigation bar with links to Terraform, Install, Tutorials, Documentation, Registry, and Try Cloud. A search bar is located in the top right corner. The main content area is divided into three sections: "Binary download", "Windows", and "Linux". The "Binary download" section provides links to download the AMD64 and ARM64 versions of Terraform 1.9.4. The "Windows" section provides a link to download the 386 version of Terraform 1.9.4. The "Linux" section provides a link to download the AMD64 version of Terraform 1.9.4. A sidebar on the left contains links to Terraform Home, Install Terraform, Operating Systems (macOS, Windows, Linux, FreeBSD, OpenBSD, Solaris), Release information, Next steps, and Resources. A footer at the bottom contains a cookie notice and an "ACCEPT" button.

Install | Terraform | HashiCorp

developer.hashicorp.com/terraform/install#windows

Search %/ctrl K

Terraform Home

Install Terraform

Operating Systems

macOS

Windows

Linux

FreeBSD

OpenBSD

Solaris

Release information

Next steps

Resources

Binary download

AMD64
Version: 1.9.4 [Download](#)

ARM64
Version: 1.9.4 [Download](#)

Windows

Binary download

386
Version: 1.9.4 [Download](#)

AMD64
Version: 1.9.4 [Download](#)

Linux

Package manager

[Ubuntu/Debian](#) [CentOS/RHEL](#) [Fedora](#) [Amazon Linux](#) [Homebrew](#)

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](#)

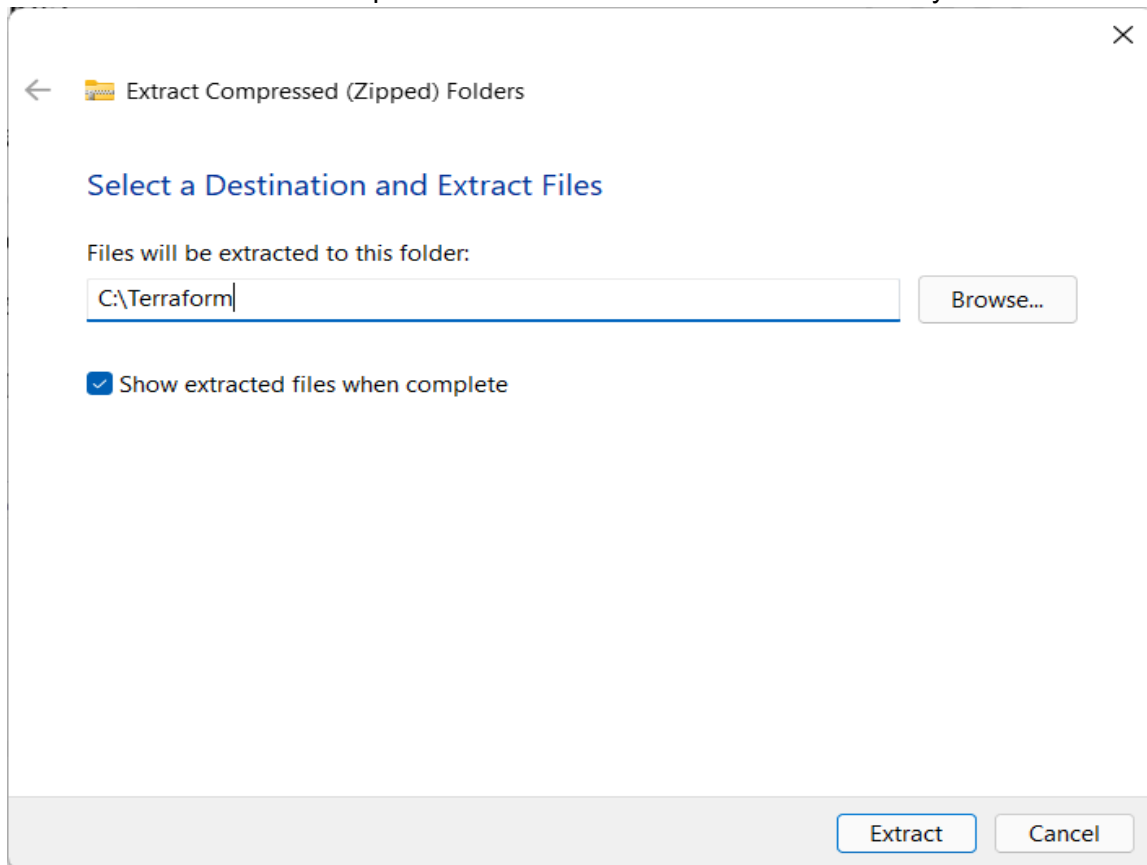
HCP Terraform

Automate your infrastructure provisioning at any scale

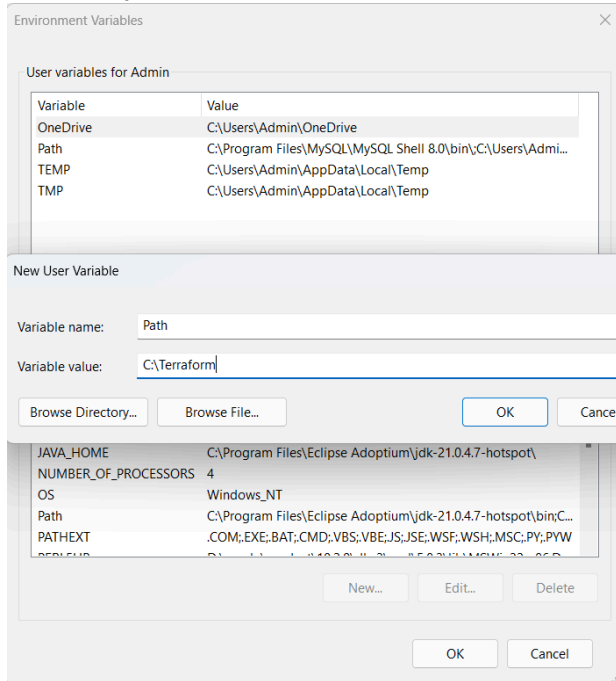
[Try HCP Terraform for free](#)

We use cookies & other similar technology to collect data to improve your experience on our site, as described in our [Privacy Policy](#) and [Cookie Policy](#). [Manage Preferences](#) [ACCEPT](#)

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



Set the System path for Terraform in Environment Variables



Open PowerShell with Admin Access and check its functionality.

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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
  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
  taint     Mark a resource instance as not fully functional
  test      Execute integration tests for Terraform modules
  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> echo AryanSaraf
AryanSaraf
PS C:\WINDOWS\system32>

```

CONCLUSION :

Terraform is a powerful tool for managing infrastructure as code, offering automation, consistency, and flexibility across various platforms. Understanding its core concepts and lifecycle is essential for effective infrastructure management.