

Practical Manual on Terraform

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high level components such as DNS entries, SaaS features, etc.

Terraform can manage infrastructure on multiple cloud platforms. Terraform's state allows you to track resource changes throughout your deployments. You can commit your configurations to version control to safely collaborate on infrastructure. Terraform plugins called providers let Terraform interact with cloud platform and other services via their application programming interfaces (APIs).

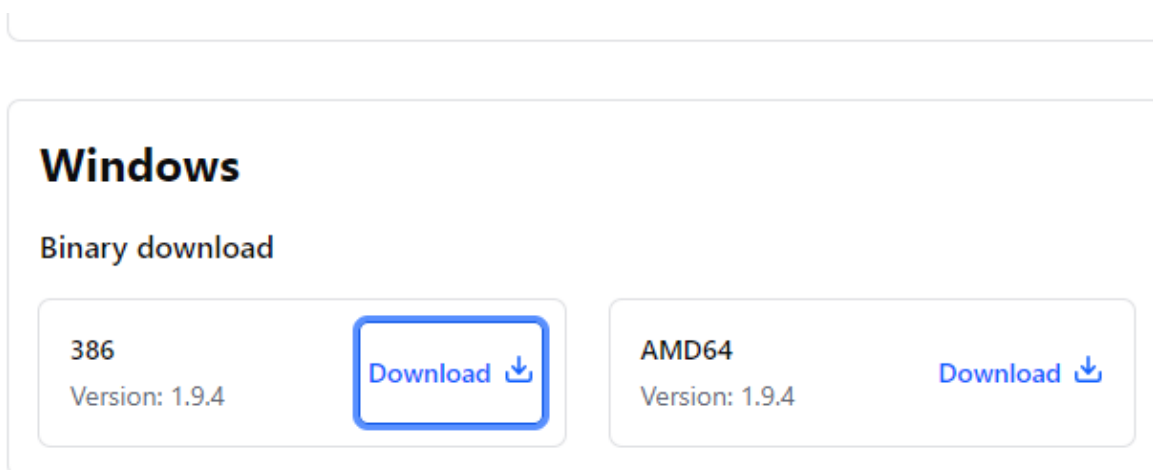
A) Installation and Configuration of Terraform in Windows

Step 1: Download terraform

To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

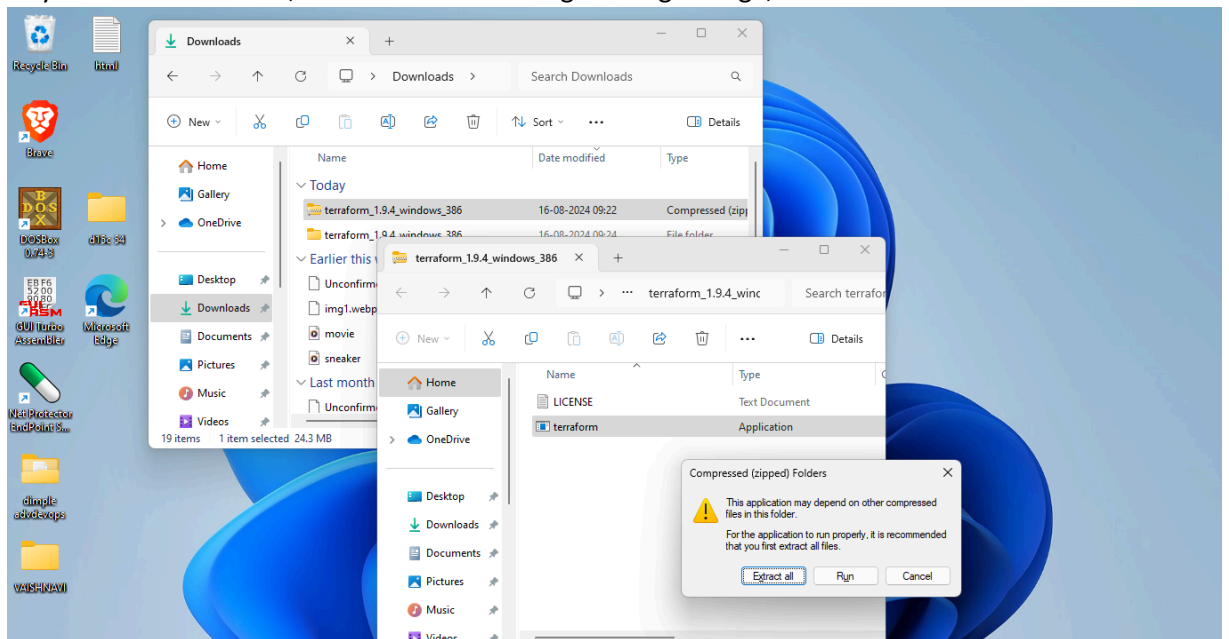
website: <https://www.terraform.io/downloads.html>

Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.

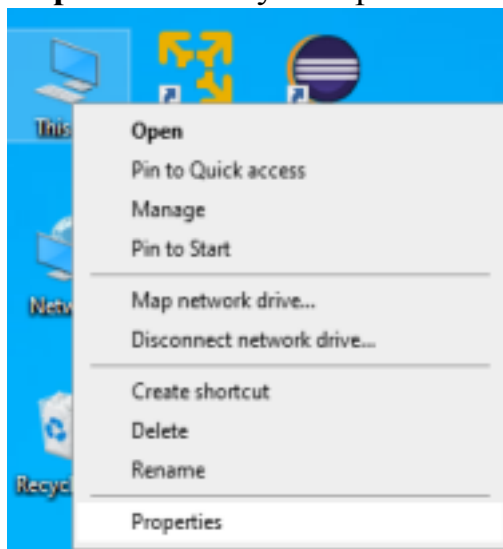


Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory

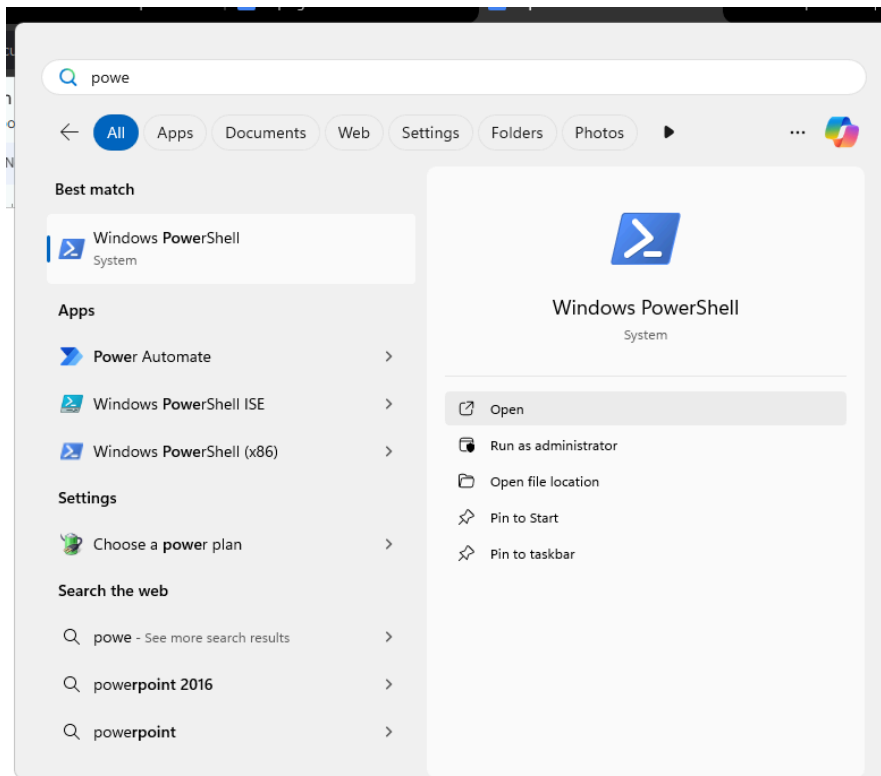
Prepared by Dr. Bhushan Jadhav, Thadomal Shahani Engineering College, #9702868662



Step 3: Set the System path for Terraform in Environment Variables



Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality

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

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

PS C:\Users\Student.VESIT505-03.000> 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:\Users\Student.VESIT505-03.000> |
```

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

