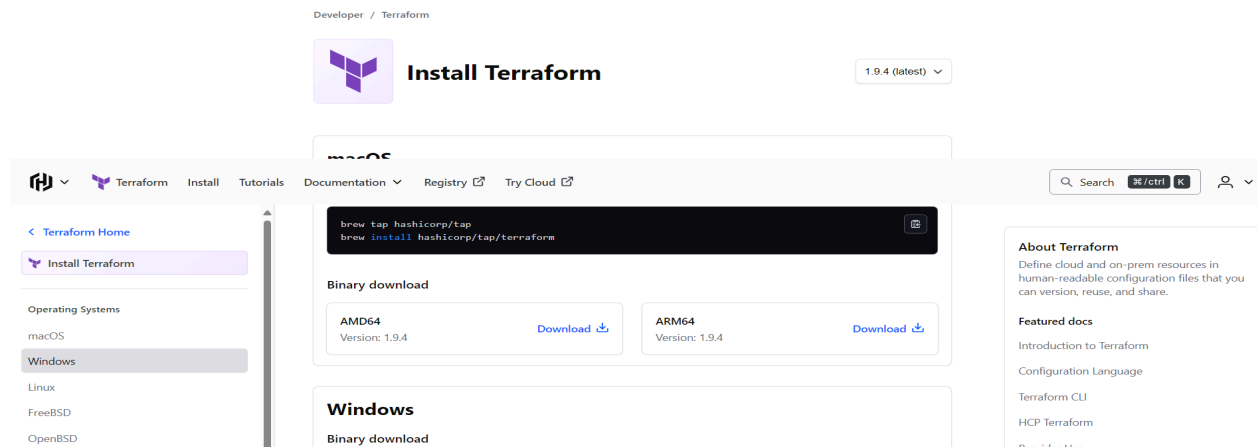
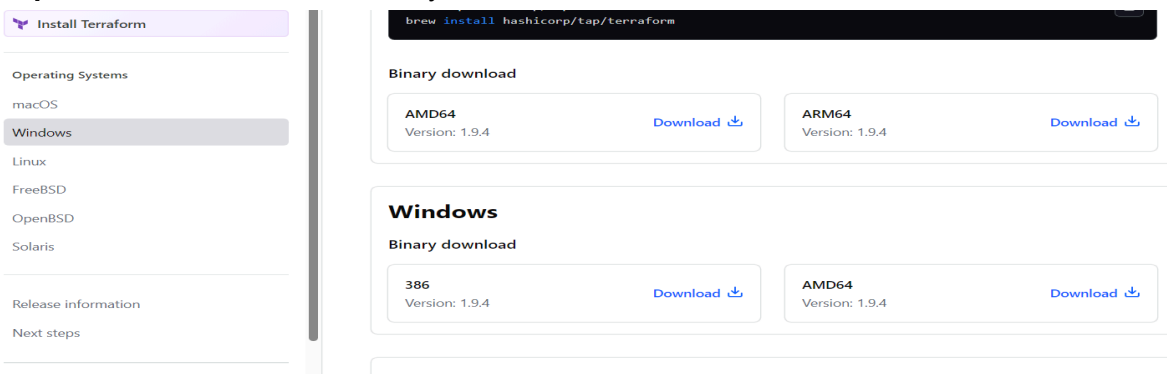


## Step 1: Download terraform

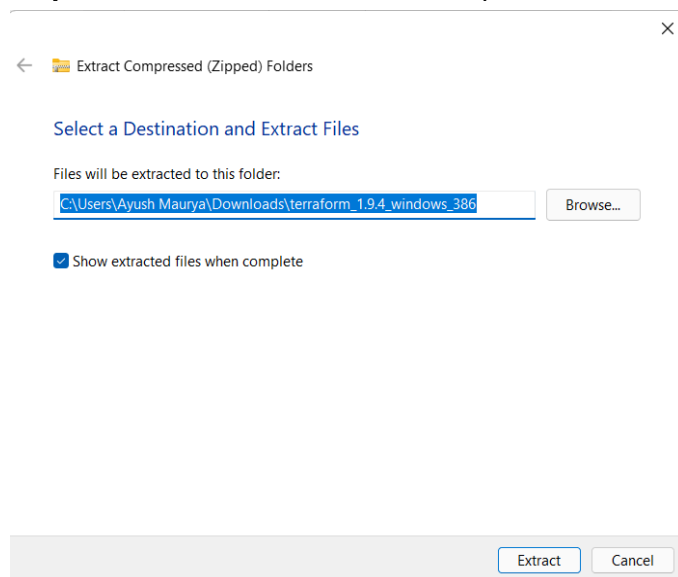
website: <https://www.terraform.io/downloads.html>



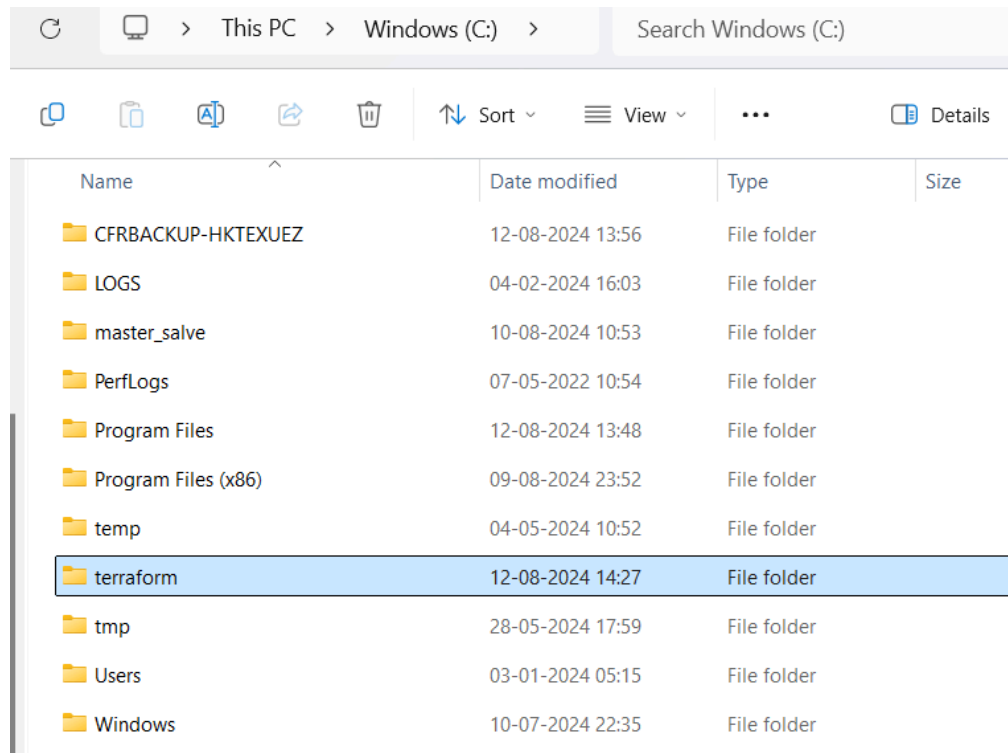
## Step 2: Download Windows Binary download 386



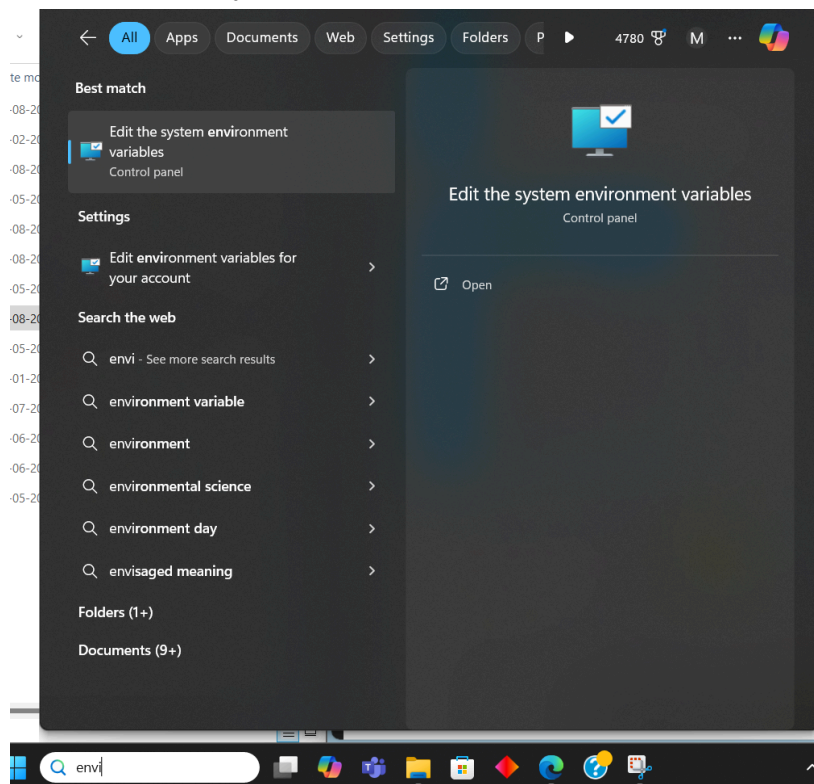
## Step 3: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



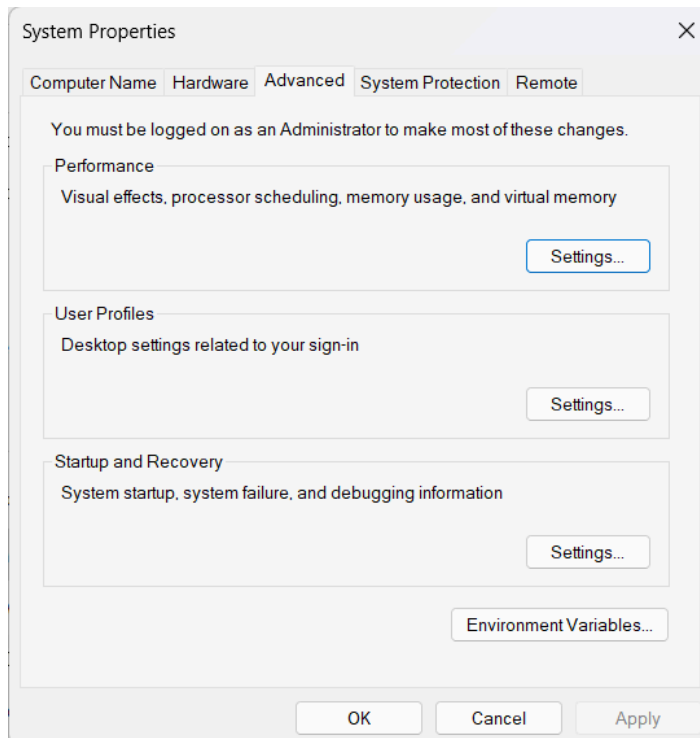
#### Step 4: Created folder in C: drive for terraform,



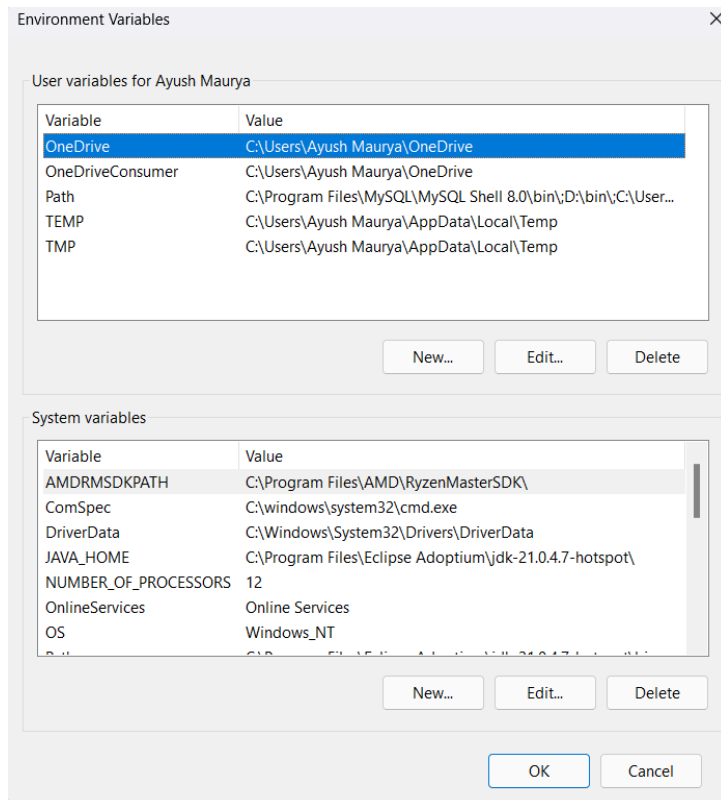
#### Step 5: Set the System path for Terraform in Environment Variable. Open Environment Variable.



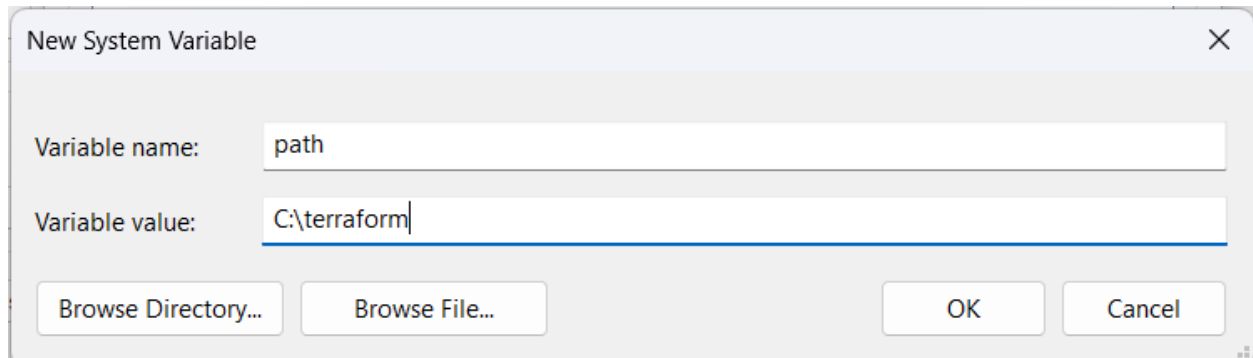
**Step 6:** System properties will open. Now click on Environment Variables On bottom.



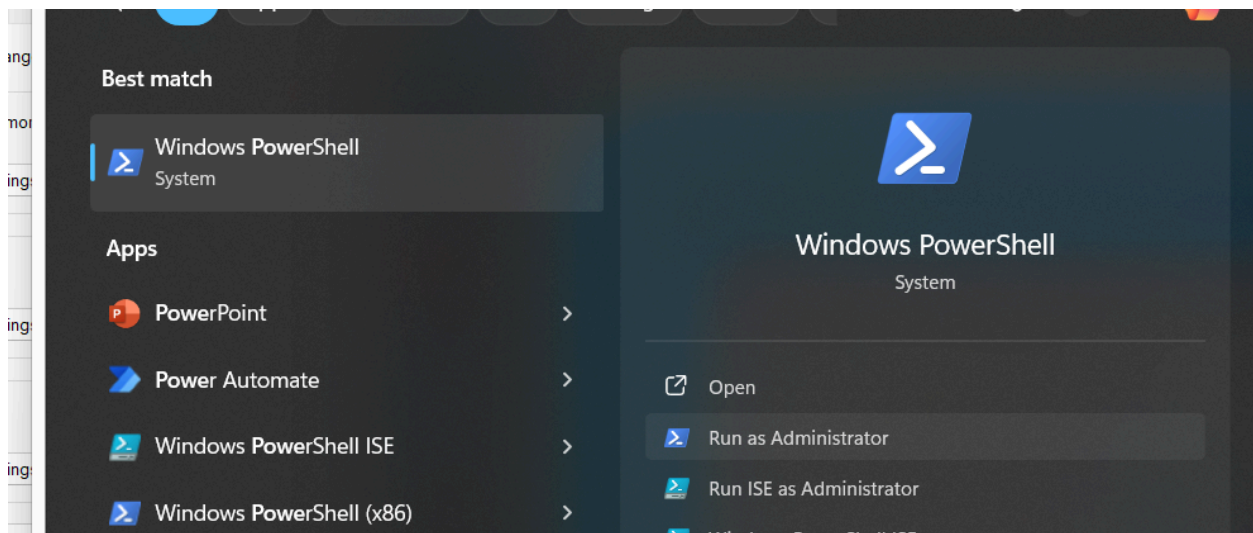
**Step 7:** Click on 'New' to create new system variables.



**Step 8:** Give variable name as 'path' or any and variable value as path of folder in C: drive where terraform is extracted.



**Step 9:** Open PowerShell with Admin Access



**Step 10:** Open Terraform in PowerShell and check its functionality

```
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.
```