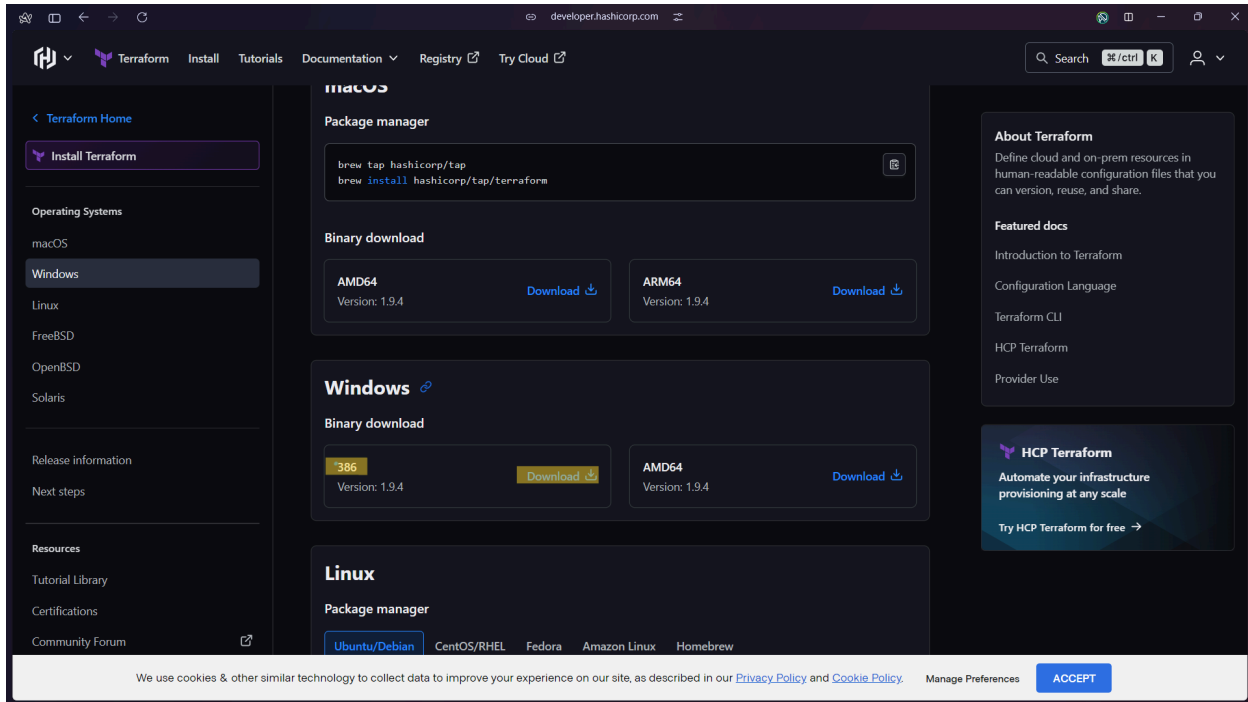
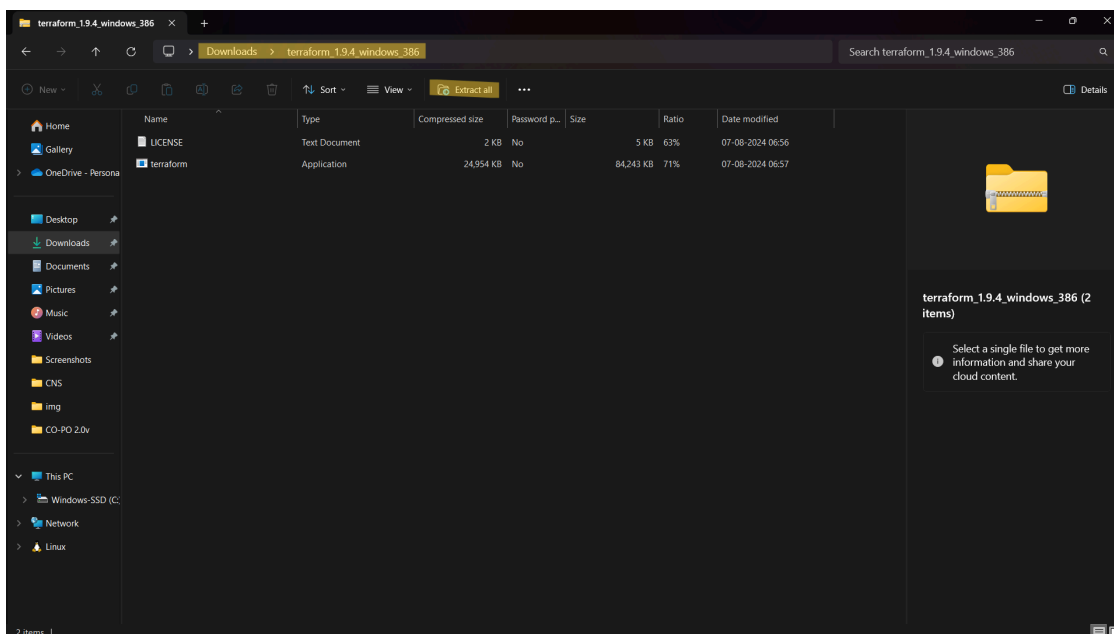


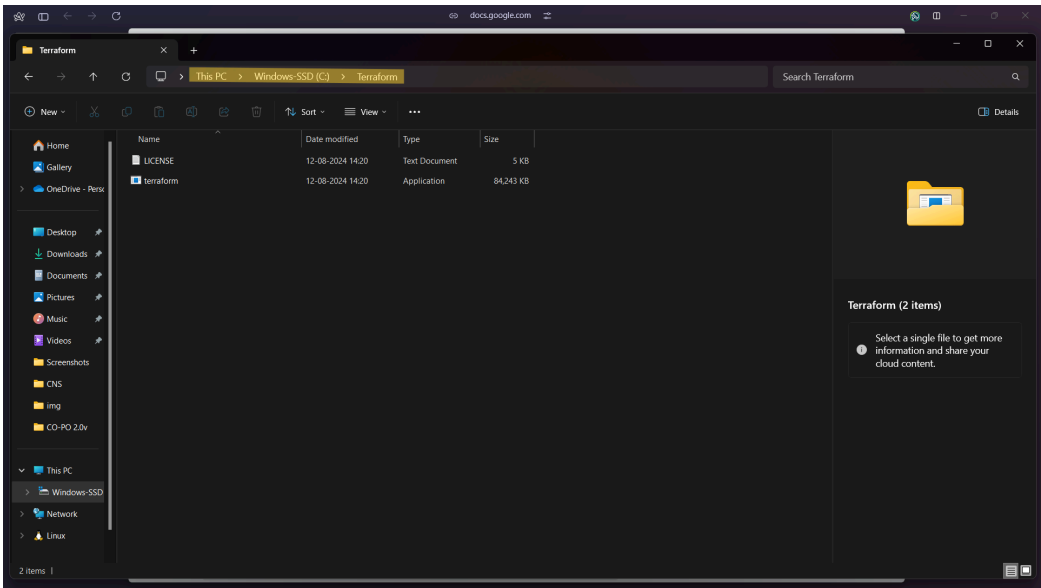
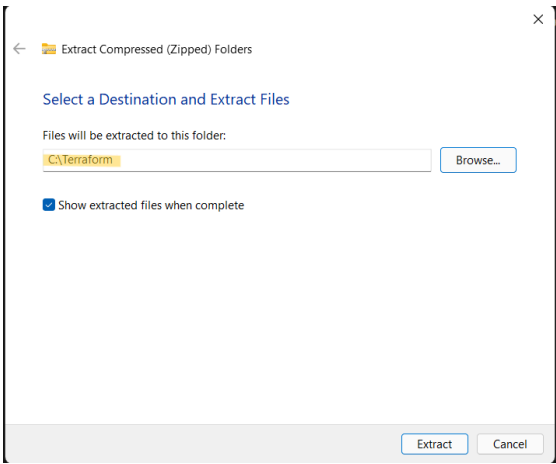
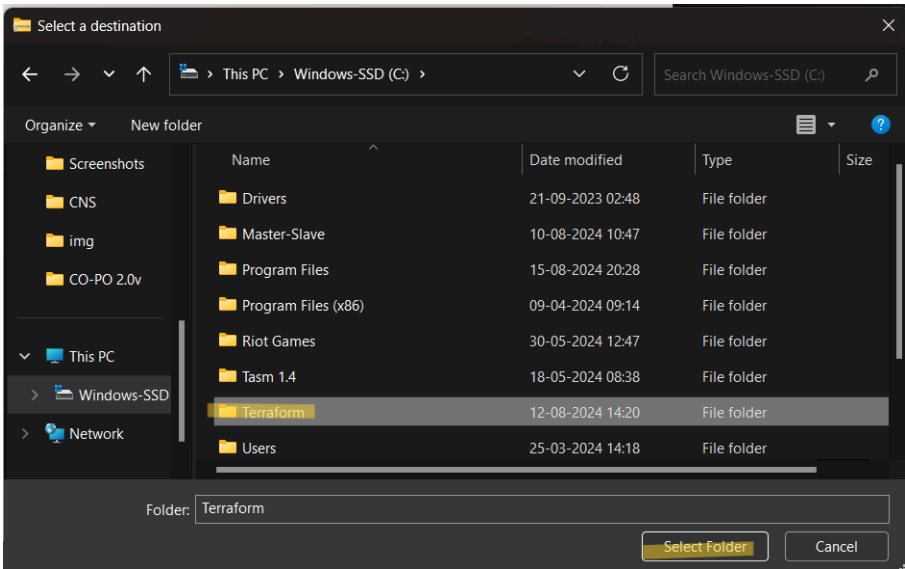
**Aim:** To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine and Windows.

**Step 1:** Visit <https://developer.hashicorp.com/terraform/install>. Scroll down to the operating system on which you want to install Terraform. For Windows, select 386 option.

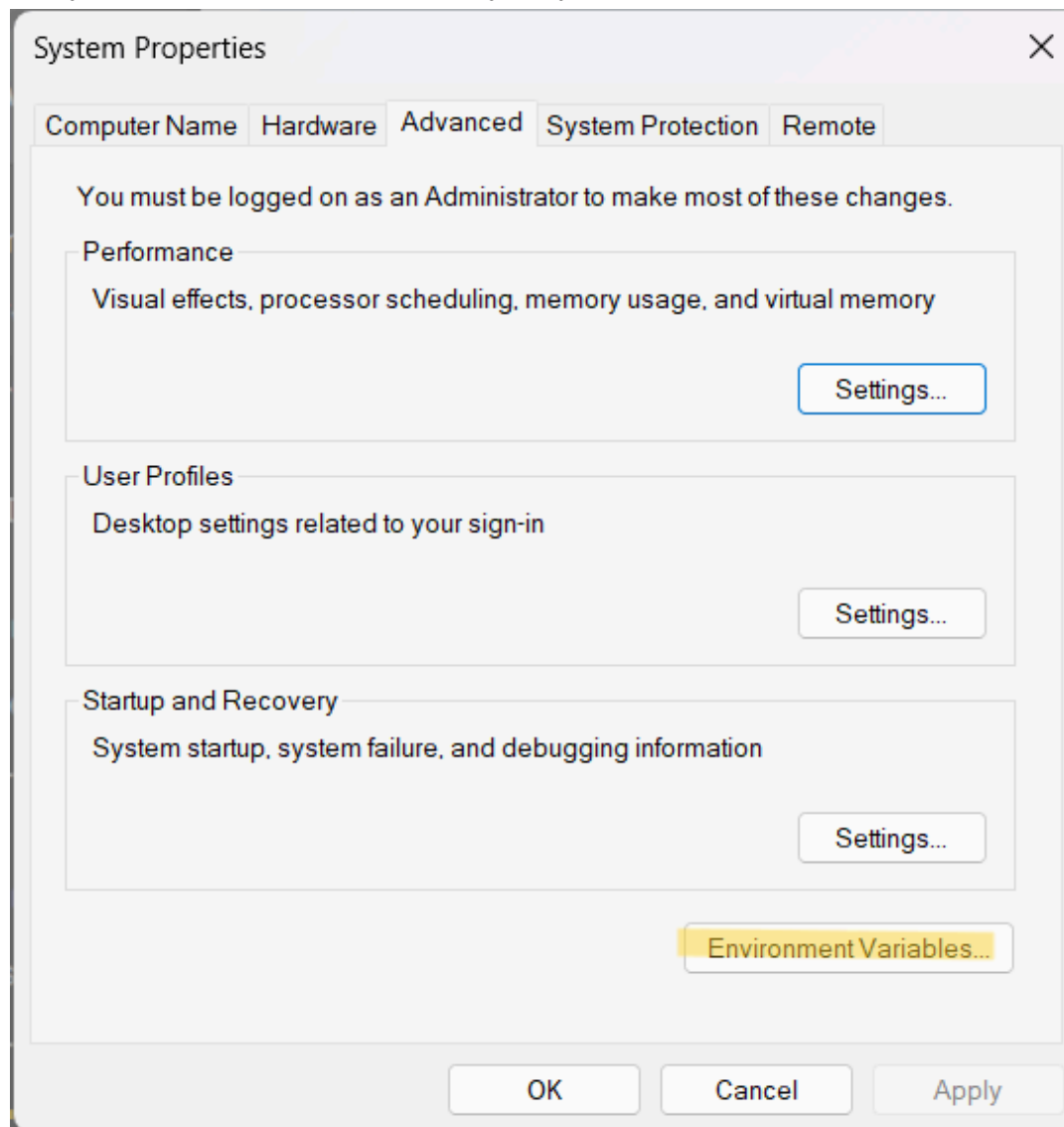


**Step 2:** A zip file gets downloaded to your system. Create a folder on your C-drive named as Terraform. Go to the zip you downloaded, click on extract all, and select the C-drive file we just created. This will extract the Folder in C:\Terraform.



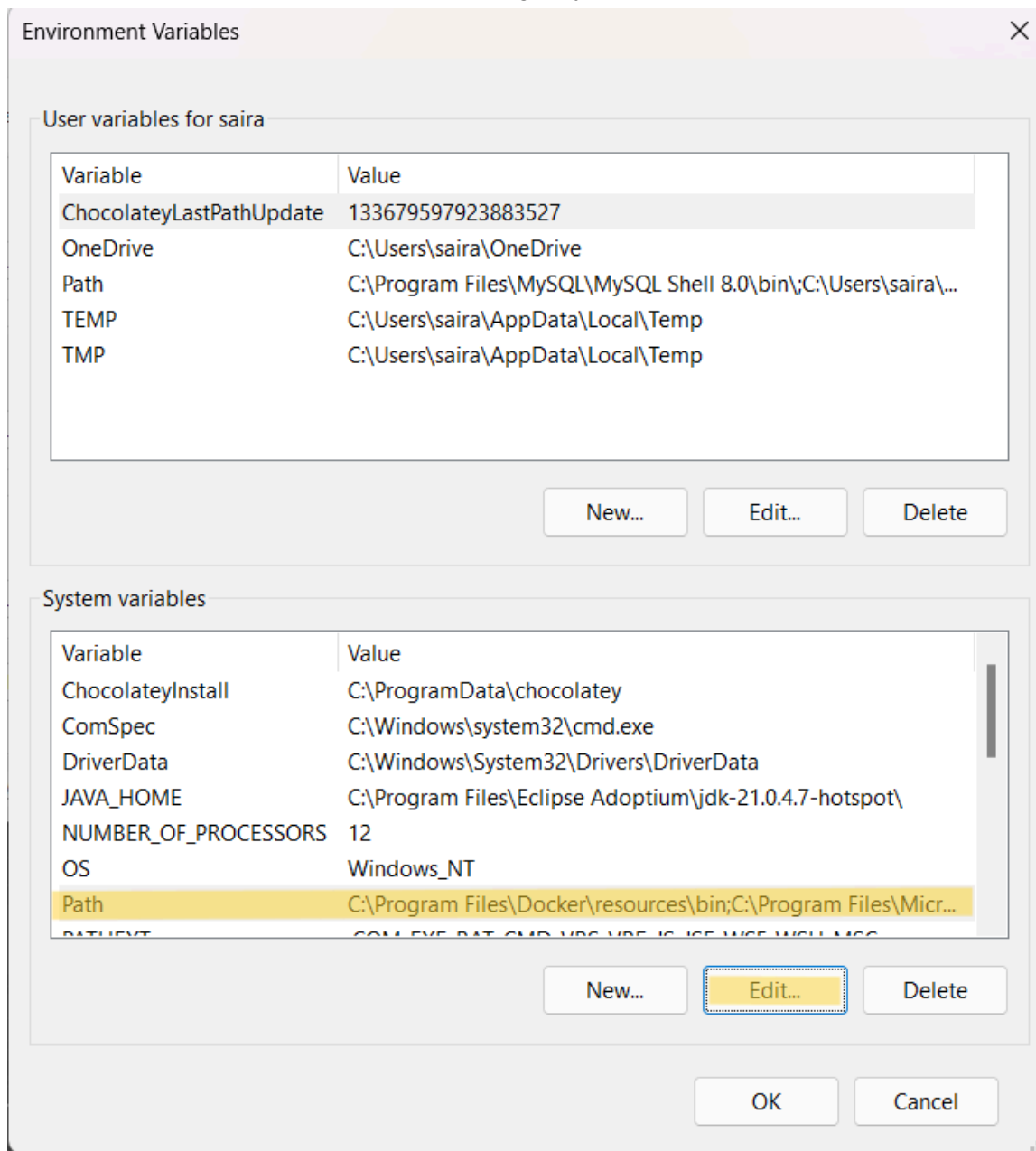


**Step 3:** After extracting the files, we need to add the folder path as a User System variable so that the commands could be used in the command prompt or powershell. For this, search Edit the system environment variables on you system. Open it, click on Environment Variables.

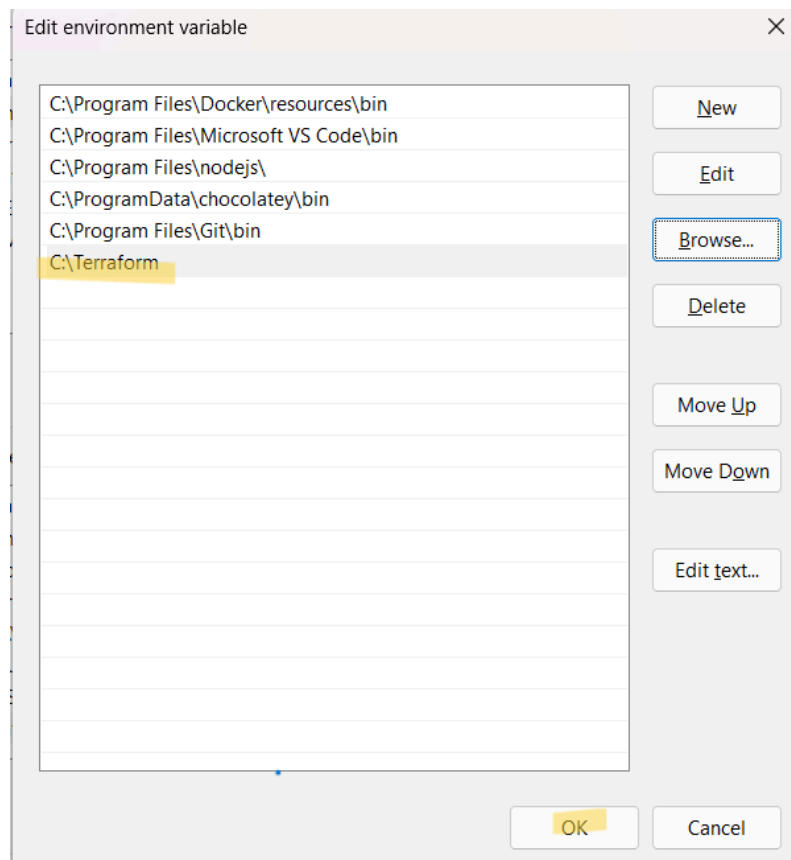
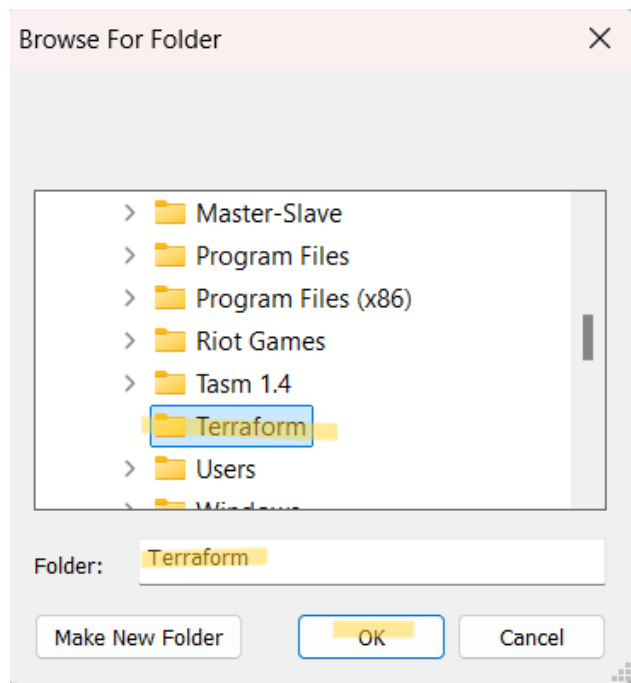


**Step 4:** This opens up an Environment Variables window. Here, under System Variables:

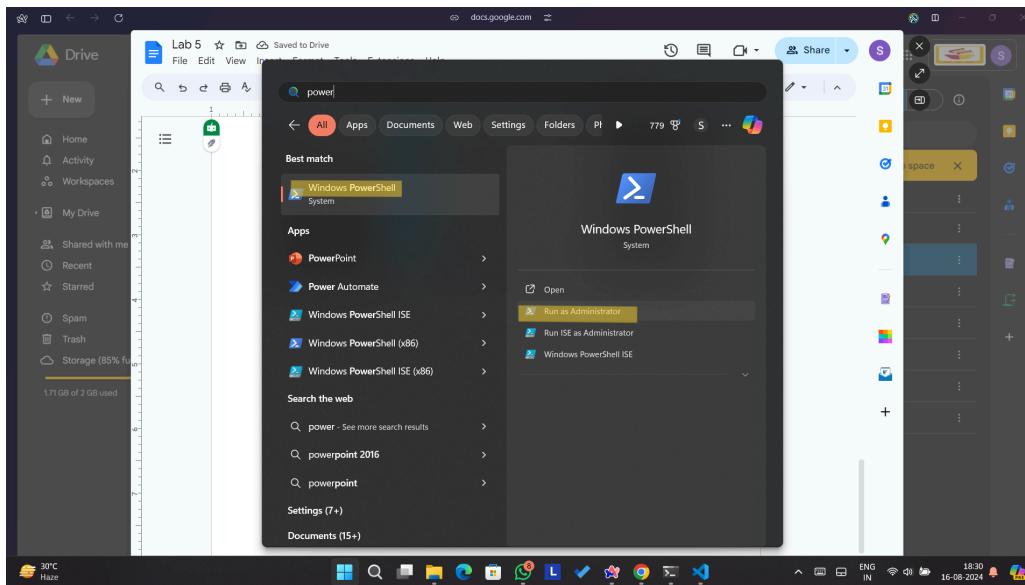
- 1) If a variable called Path does not exist, create the variable.
- 2) If Path variable exists, click on it; this will give you an option to edit the variable. Click on Edit.



**Step 5:** This gives you a screen to add a path to the existing paths. Click on Browse, search for the Terraform folder (C:\Terraform) and click on Ok. This adds the link and it can be seen in the list of paths. Click on Ok.



**Step 6:** Click on OK until you exit from all the Environment Variable screens. Now go to your search and search for Windows Powershell. Run this as an administrator.



**Step 7:** As the powershell opens, run the command terraform. This would give you all the main commands, other commands, and global options. If this is your output, then you have successfully installed Terraform on your system.

```
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.
```