Mayuri Shindalkar
D15B - 70

PAGE NO.:
DATE / /

Advance DevOps

EXPERIMENT - 5

AIM: To understand terraform lifecycle, core concepts / terminologies and install it on a linux machine.

THEORY: Terraform is an infrastructure as code 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 both existing service providers and custom in-house solutions.

* Key features

- Infrastructure as Code: We describe our infrastructure using terraform's high-level configuration language in human-readable declarative configuration files. This allows you to create a blueprint that you can version, share and reuse.

- Resource Graph: Terraform builds a resource graph and creates or modifies non-dependent resources in parallel. This allows Terraform to build resources as efficiently as possible and gives you greater insight in to your infrastructure.

- Change automation: Terraform can apply complex changesets to your infrastructure with minimal human interaction. When you update configuration files, Terraform determines what changed and creates incremental execution plans that respect dependencies.

PAGE NO.:
DATE / /

- Terraform actually works, there's sort of two major components:
One is the terraform core: It takes the terraform configuration which is being provided by the user. and then takes the terraform state which is managed by terraform itself. As such, this gets fed into the core that is responsible for figuring out what is that graph of our different resources for example how these different pieces relate to each other or what needs to be created/updated/destroyed. It does all the essential lifecycle management.

- On the backside, terraform supports many different providers such as: cloud providers and they also could be on-premise infrastructure. But this support is not restricted or limited only to infrastructure. As a service, terraform can also manage higher level like platform-As A service or even software (kubernetes) (Lambdas). As A service (Datadog, Github)

- All these are important pieces of the infrastructure, they are all part of logical end to end delivery.

- Terraform has over a hundred providers for different technologies and each provider gives terraform users access to their resources. It also gives you the ability to create infrastructure at different levels.

PAGE NO.:
DATE / /

⇒ Terraform Core Concepts:

- Variables: Also used as Input variables, it is a key-value pair used to by Terraform modules to allow customization.
- Provider: It is a plugin to interact with APIs of service and access its related resources.
- Module: It is a folder with terraform templates where all the configurations are defined.
- State: It consists of cached information about Infrastructure managed by terraform and its related configurations.
- Resources: It refers to a block of one or more Infrastructure objects which are used in configuring and managing the Infrastructure.
- Data source: It is implemented by providers to return information on external objects to terraform.
- output values: These are return values of a terraform module that can be used by other configurations.

- Plan : It is one of the stages where its determining what needs to be created, updated or destroyed to move from real state of infrastructure to the desired state.

- Apply : It is one of the stages where it applies the changes in real state of infrastructure in order to move to desired state.

Conclusion : We successfully understood terraform lifecycle, core concepts and installed it on linux machine.

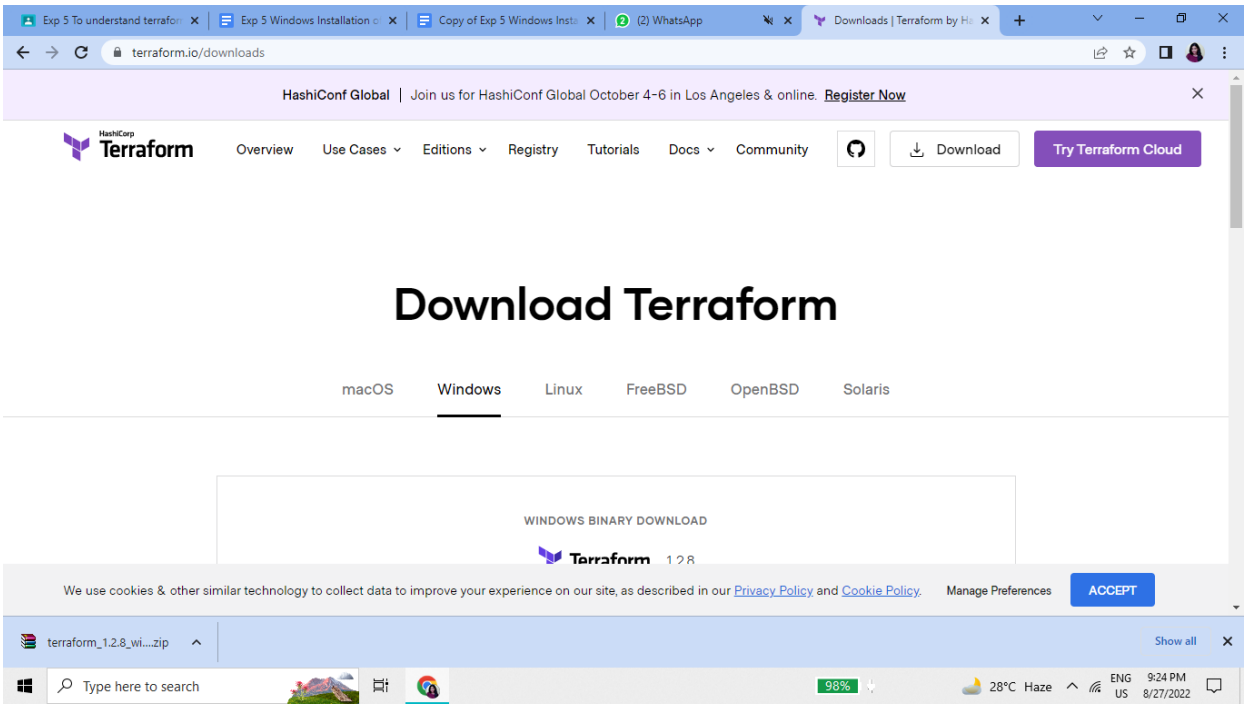
Implementation:-

A) Installation and Configuration of Terraform in Windows

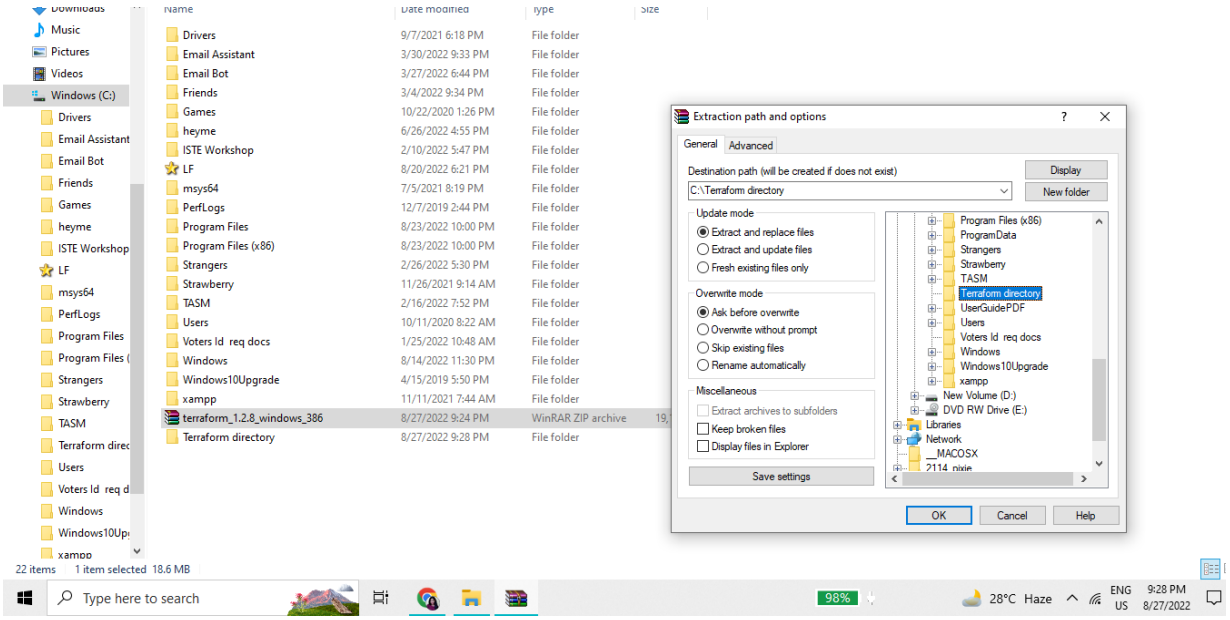
Step 1: Download terraform

To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website:-
https://www.terraform.io/downloads

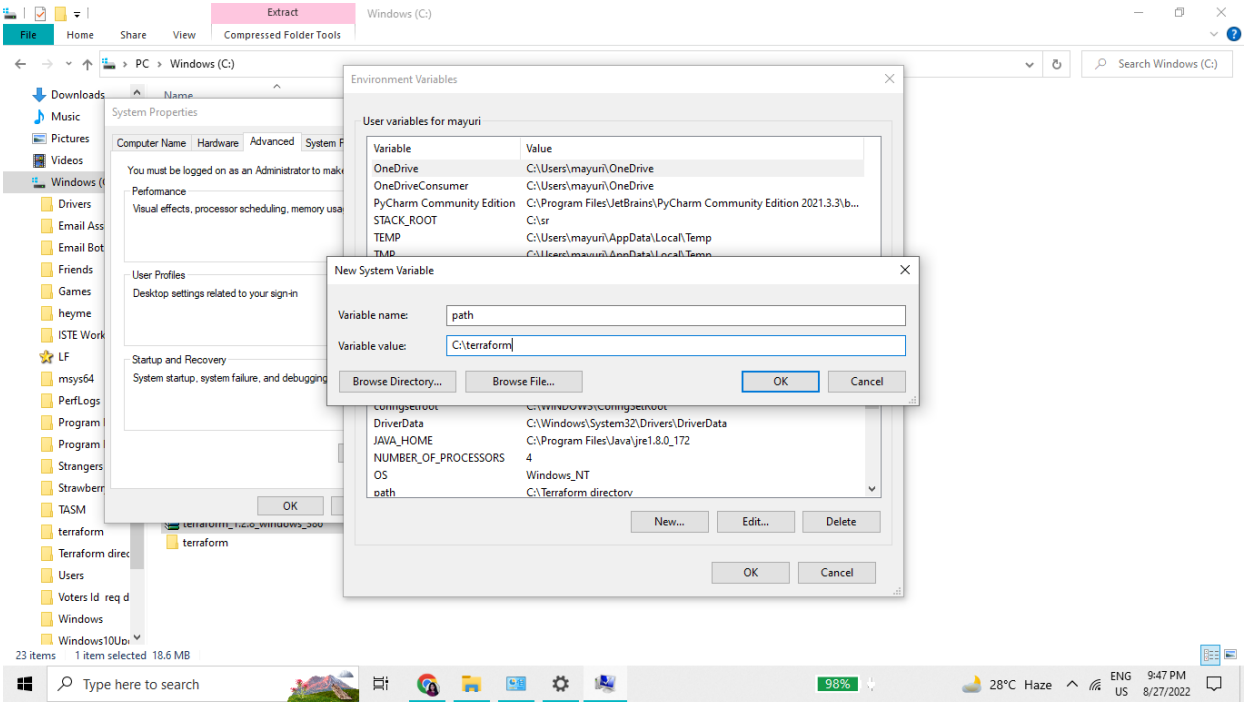
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.



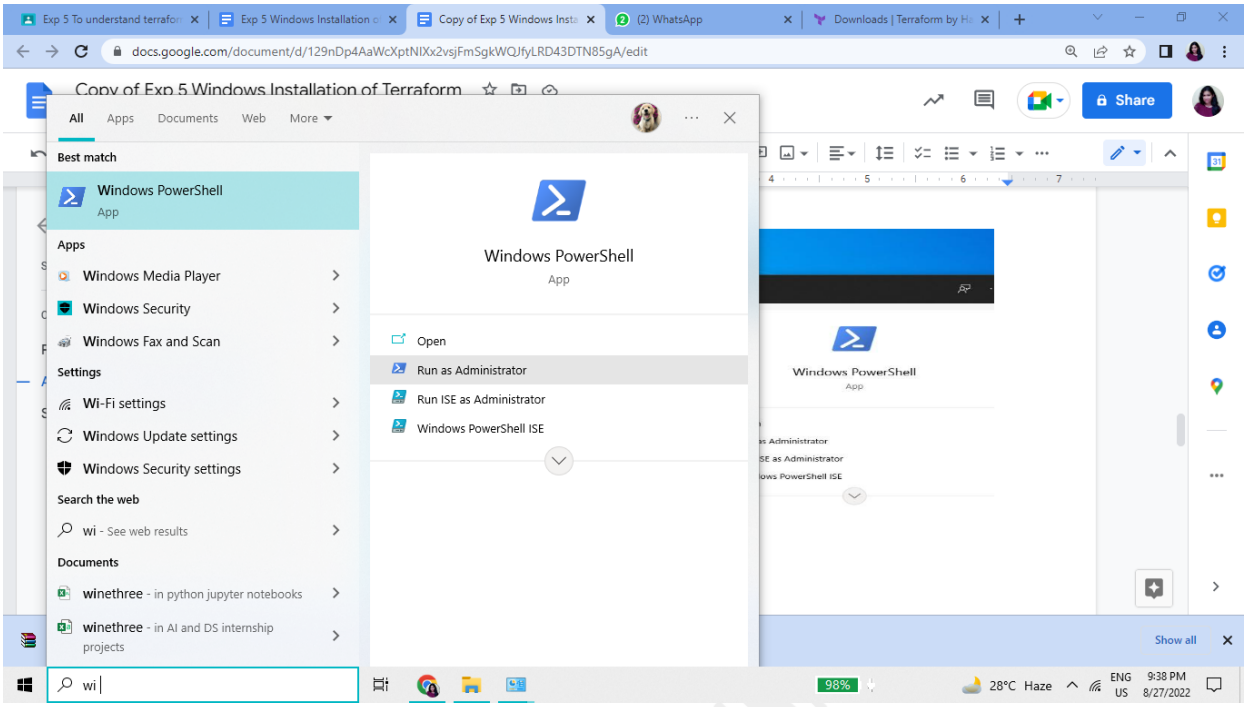
Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



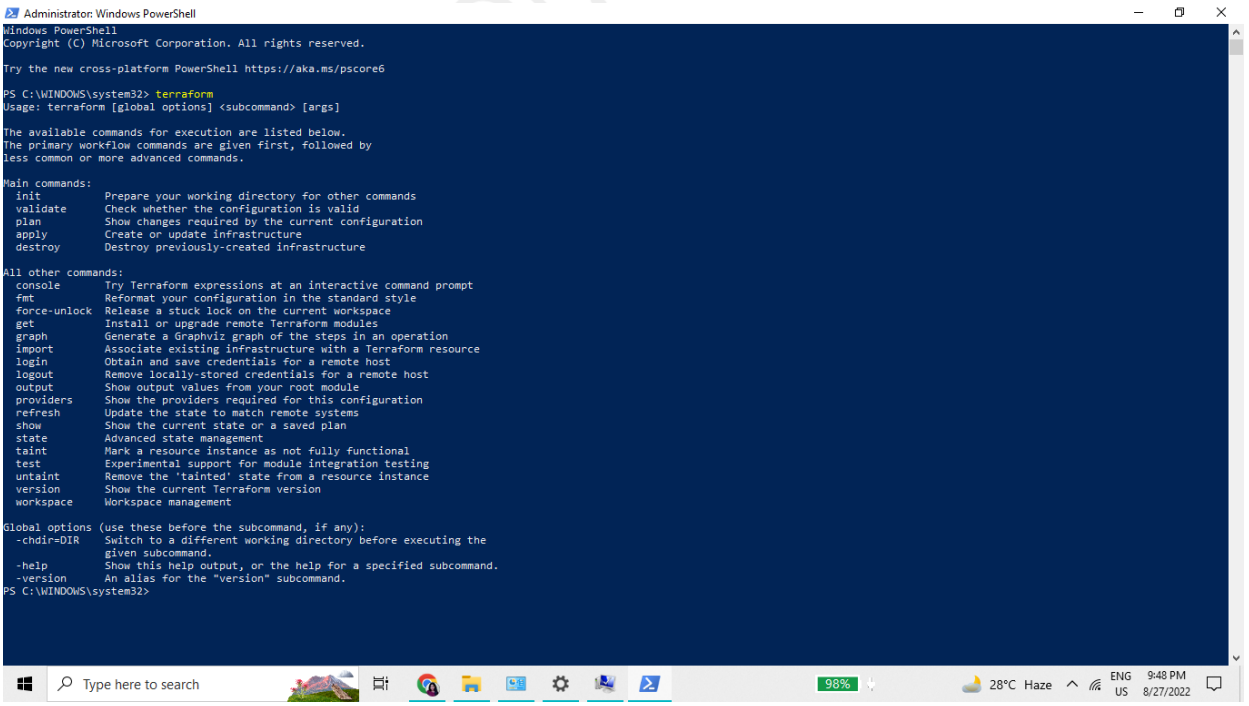
Step 3: Set the System path for Terraform in Environment Variables

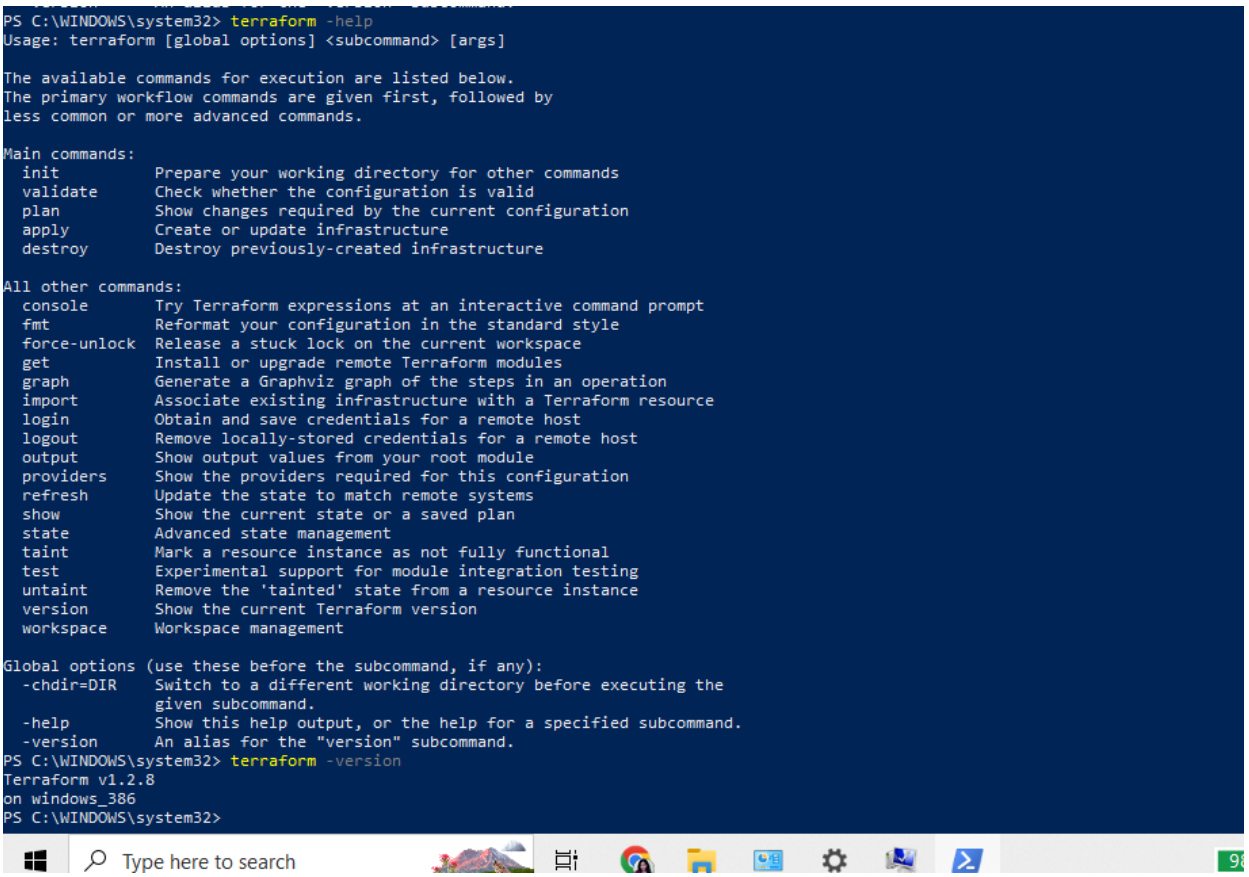


Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality





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

Conclusion: We successfully understood terraform lifecycle, core concepts/terminologies and installed it on a Linux Machine.