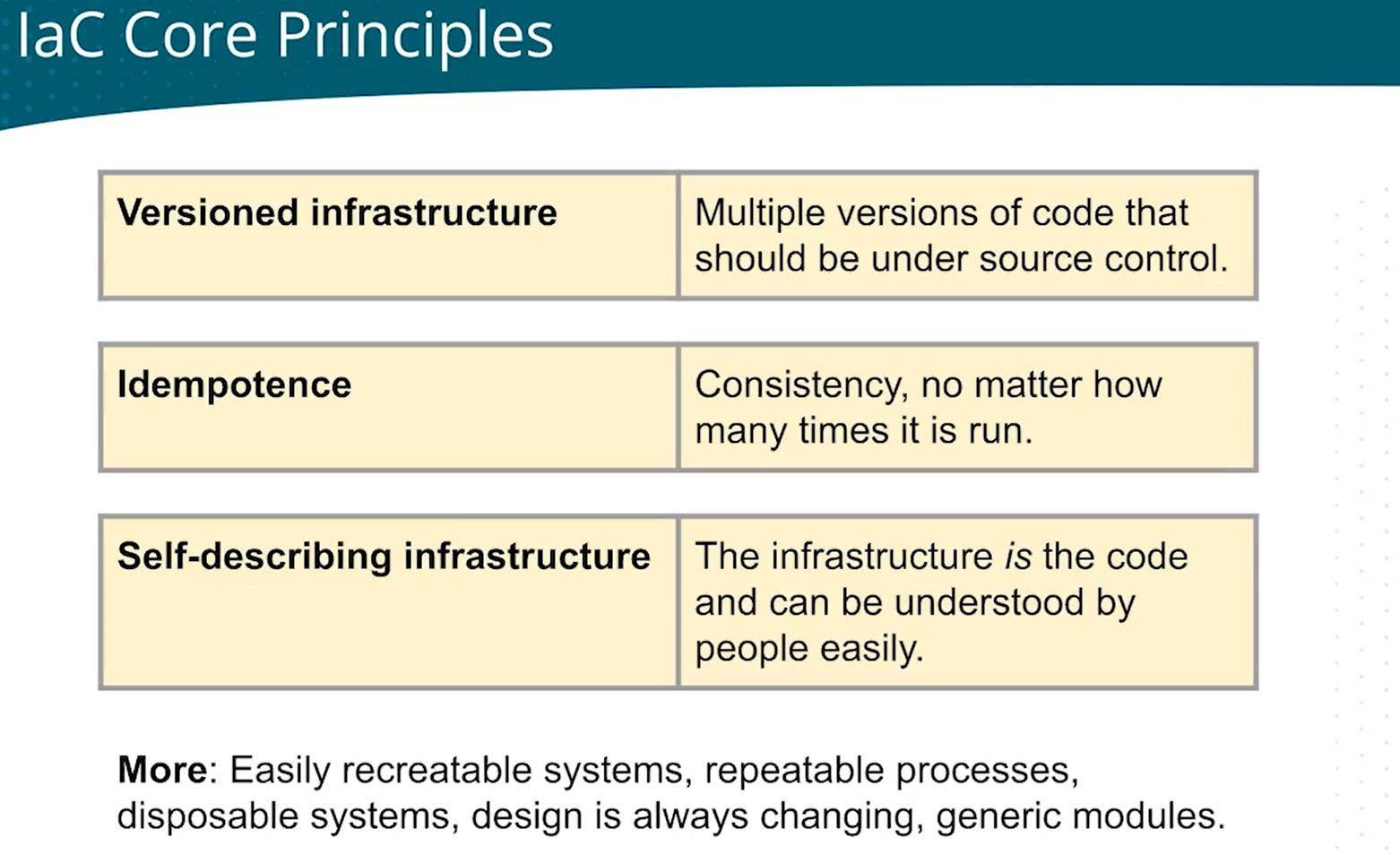


Terraform is a declarative coding.



Terraform is designed to be idempotent. Idempotence means that performing an operation multiple times has the same effect as performing it just once. In the context of Terraform, this means that if you apply a configuration multiple times, it should result in the same desired state without causing any unintended changes to the infrastructure.

Here's how Terraform achieves idempotence:

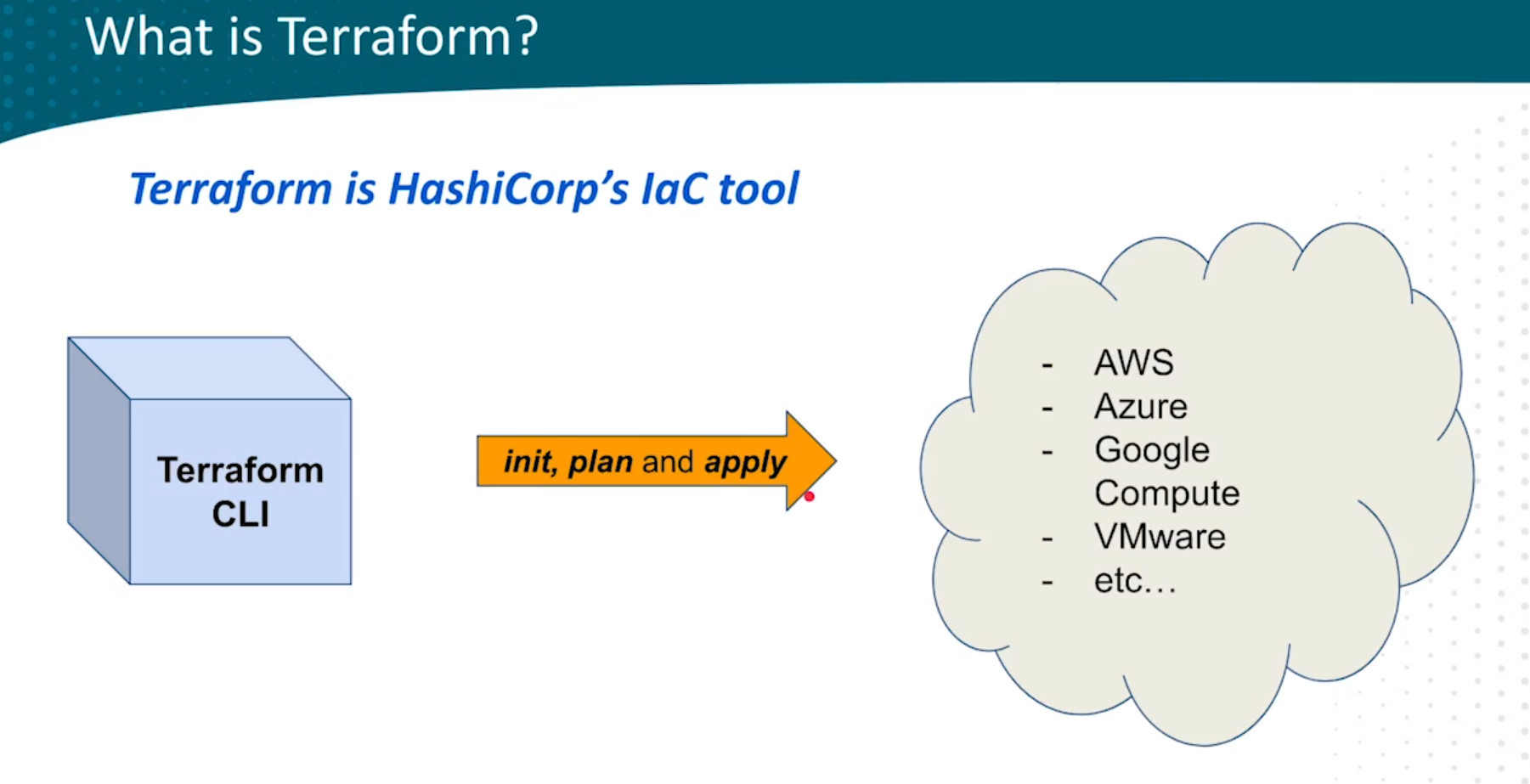
1. \*\*State Management:\*\* Terraform keeps track of the current state of your infrastructure in a state file. When you run `terraform apply`, Terraform compares the current state with the desired state described in your configuration files.

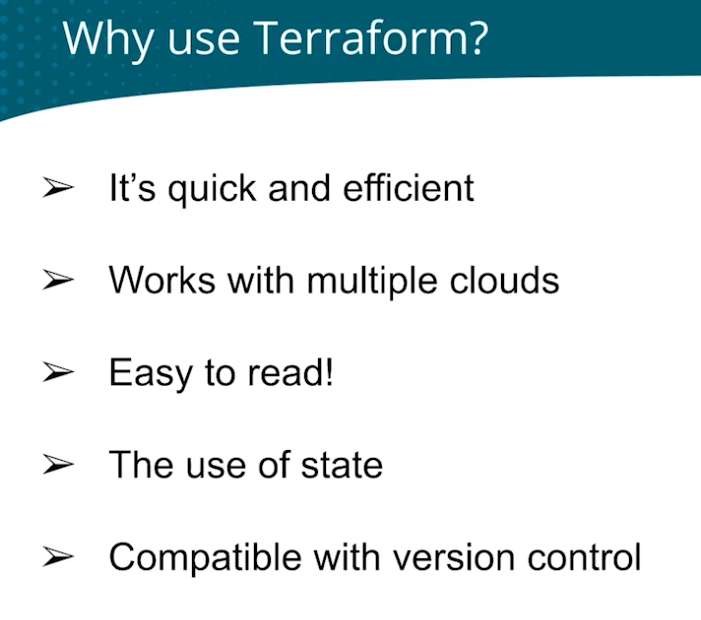
2. \*\*Planning:\*\* Before making any changes, Terraform generates a plan that outlines what actions will be taken to reach the desired state. This plan is reviewed and approved by the user before applying any changes.

3. \*\*Differential Analysis:\*\* Terraform performs a differential analysis to determine what changes need to be made. It only applies the changes necessary to bring the infrastructure to the desired state, ignoring any redundant or identical changes.

4. \*\*Declarative Nature:\*\* Terraform configurations are declarative, meaning you describe the desired state of your infrastructure rather than a series of commands to achieve that state. This declarative approach helps ensure that the final state of the infrastructure is consistent with the configuration, regardless of the number of times you apply it.

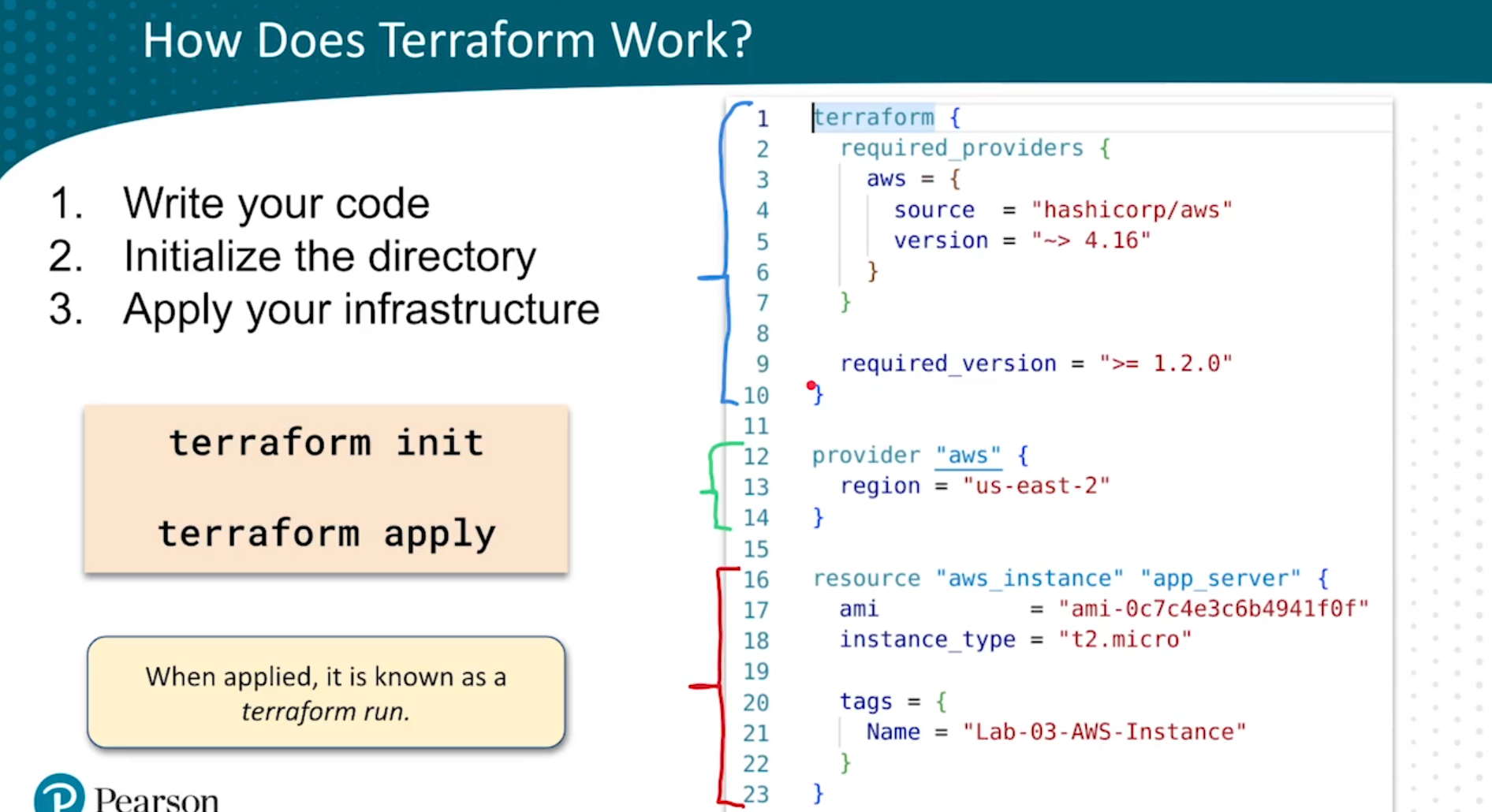
By combining these features, Terraform ensures that applying the same configuration multiple times does not result in any unintended changes, making it idempotent.





**Terraform -help or terraform -h** 🡪 used for help of know about any commands.

**If you want any help in subcommand like init 🡪 terraform -h init**



Here what is happening is terraform is talking to AWS.

We first download the plugin 🡪 then terraform can communicate to AWS API

Note: - terraform init is the command we use to download the plugin.

We declare what provider is it in green section.

In red section we are telling what action needs to be done.

<BLOCK TYPE> <BLOCK LABEL> <BLOCK LABEL NAME>

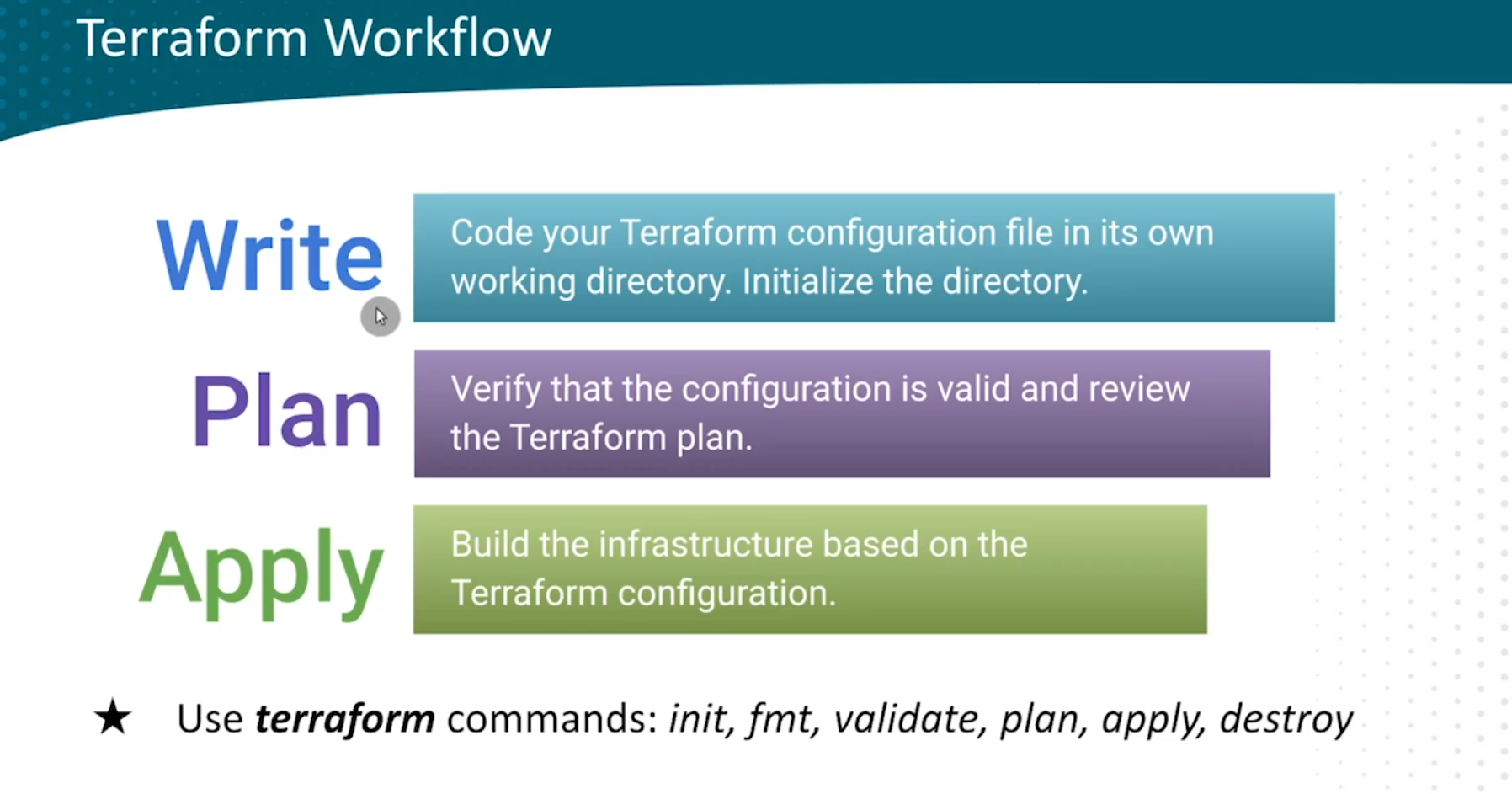
resource is the block type.

aws\_instance is the block label.

app\_server is the block label name.

tag section is the AWS tag that we give.

**NOTE: - ALWAYS USE TERRAFORM OFFICIAL DOCUMENTATION FOR ANY DOUBTS.**

****

**Terraform fmt 🡪 it is used to format the code.**

**Terraform validate 🡪 validate the errors.**