**Terraform Variables (Detailed Coverage)**

**1. Syntax**

* **Block:**
* variable "name" {
* type = string
* default = "value"
* description = "explanation"
* }
* **Labels:** variable (block type), "name" (label).
* **Identifiers:** Names for variables/resources.
* **Expressions:** Assign values (=, ${var.name}).
* **Comments:** # single line, /\* \*/ multi-line.

**2. Variables**

* **Input Variables** → provide values dynamically.
* **Output Variables** → display values after apply.
* output "instance\_ip" {
* value = aws\_instance.myec2.public\_ip
* }

**3. JSON Format**

* Equivalent of .tf in JSON (.tf.json).
* {
* "variable": {
* "region": {
* "type": "string",
* "default": "us-east-1"
* }
* }
* }

**4. Multiple Blocks**

* Can define multiple variable blocks in a file.
* Example:
* variable "region" {}
* variable "instance\_type" {}

**5. Input Variables**

**Ways to provide values:**

1. **Default in variable block**
2. **Command line** → terraform apply -var="region=us-east-1"
3. **.tfvars file** (e.g. terraform.tfvars)
4. region = "us-east-1"
5. instance\_type = "t2.micro"
6. **Multiple .tfvars files** → terraform apply -var-file="dev.tfvars"
7. **Environment variables** → export TF\_VAR\_region=us-east-1

**6. Multiple Input Variables**

* Supported, passed together via .tfvars or command line.

**7. Variable Types**

* string → "us-east-1"
* number → 3
* bool → true/false
* list → ["t2.micro","t2.small"]
* map → { env = "dev", region = "us-east-1" }
* object → { name = string, age = number }
* tuple → [ "aws", 3, true ]

**8. Changes in Original Configuration File**

* If variable definition changes:
  + Terraform **plans difference**.
  + May cause resource **recreation** if variable is tied to immutable property.
  + Example: changing instance\_type → replaces EC2.