Terraform: Interact with Terraform modules

- **Section 5 : Interact with Terraform modules**
  - Contrast Module Source Options
  - Interact with Module Inputs and Outputs
  - Describe Variable Scope within Modules
  - Discover Modules from the Public Terraform Module Registry
  - Defining module version

- **Contrast Module Source Options :**
- Modules are used to Organise Configuration in Terraform.
- Modules make it easier to **navigate**, **understand**, and **update** your configuration by keeping related parts of your configuration together.
- Another benefit of using modules is to **encapsulate** configuration into distinct logical components.
- **Code reusability** is the Sole feature of modules in Terraform.

- **Interact with Module Inputs and Outputs :**
- Pattern to define Input variable for module is similar to define input for terraform configuration file.

```
variable "vpc_name" {
  description = "Name of VPC"
  type        = string
  default     = "example-vpc"
}
```

- Modules also have output values, which are defined within the module with the output keyword.
- User can access them by referring to module.<MODULE NAME>.<OUTPUT NAME>.

```
output "vpc_public_subnets" {
  description = "IDs of the VPC's public subnets"
  value       = module.vpc.public_subnets
}
```

- **Describe Variable Scope within Modules :**
- Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.
- **Root Module** - Every Terraform configuration has at least one module, known as its root module.
- **Child Module** - A Terraform module can call other modules to include their resources into the configuration. A module that has been called by another module is often referred to as a child module.

Will see you in Next Lecture…

See you in next lecture …