**Terraform Local Values (locals)**

In Terraform, a **local value** (commonly referred to as locals) is used to define a value that you want to reuse multiple times within a configuration. Local values are especially useful for simplifying expressions and organizing your code, particularly when dealing with complex logic or calculations.

**Syntax for Defining Locals**

locals {

name = "my\_resource"

region = "us-east-1"

}

Here, name and region are local values that can be referenced throughout your Terraform configuration.

**Using Locals**

You can reference a local value using the local keyword. Example:

resource "azurerm\_resource\_group" "example" {

name = local.name # Using the local value for the name

location = local.region # Using the local value for the region

}

**Example with Conditional Logic**

To conditionally create resources in lower environments:

variable "environment" {

type = string

default = "dev"

}

locals {

allow\_resource\_creation = contains(["dev", "test", "qa"], var.environment)

}

resource "azurerm\_resource\_group" "example" {

count = local.allow\_resource\_creation ? 1 : 0

name = "rg-${var.environment}"

location = "East US"

}

**Breakdown:**

* **Local Value**: local.allow\_resource\_creation checks if var.environment is one of the lower environments.
* **Count**: Uses the local value to either create (count = 1) or skip (count = 0) the resource.

**More Complex Example**

locals {

base\_name = "resource"

environment = var.environment

resource\_name = "${local.base\_name}-${local.environment}"

should\_create = contains(["dev", "test"], local.environment)

}

resource "azurerm\_storage\_account" "example" {

count = local.should\_create ? 1 : 0

name = local.resource\_name

location = "East US"

resource\_group\_name = azurerm\_resource\_group.example.name

account\_tier = "Standard"

account\_replication\_type = "LRS"

}

**Key Points**

* locals are calculated **once during the plan phase**.
* They can be used throughout the configuration.
* Local values improve **code reuse**, **readability**, and **maintainability**.
* You can use **expressions** and **complex logic** in locals to simplify repeated patterns.