

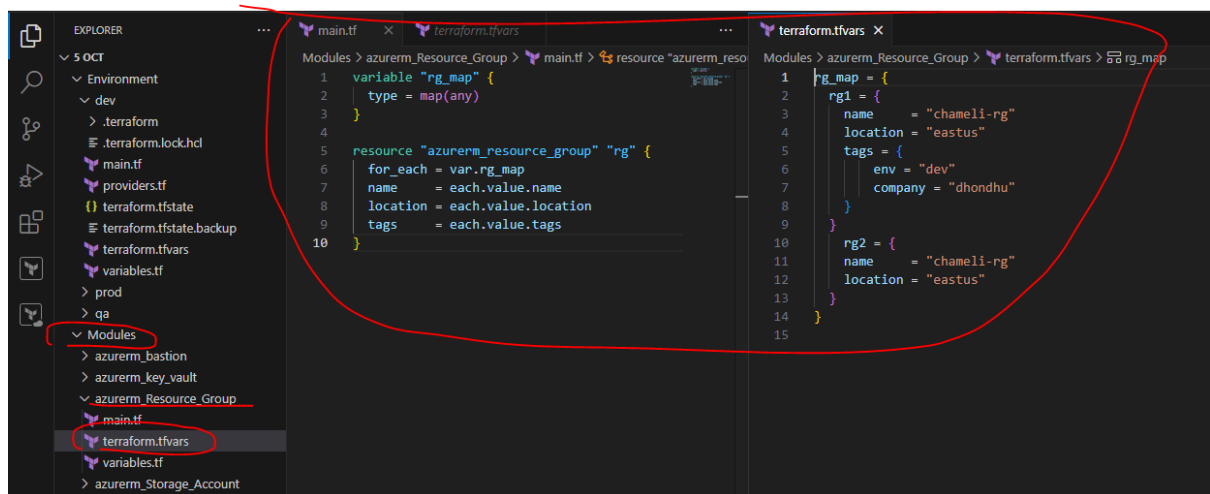
5 OCTOBER

- 1) What is logic app?
- 2) What is Azure function?
- 3)

+++++

### AGENDA – Dynamic block for rg or Using optional attribute

- 1) Adding tags in one and not in another



- 2) Arguments that can be included in rg are

```
resource "azurerm_resource_group" "example" {  
  name      = "example"  
  location = "West Europe"  
}
```

Copy

## Arguments Reference

---

The following arguments are supported:

- location - (Required) The Azure Region where the Resource Group should exist.  
Changing this forces a new Resource Group to be created.
  - name - (Required) The Name which should be used for this Resource Group. Changing this forces a new Resource Group to be created.
- 
- managed\_by - (Optional) The ID of the resource or application that manages this Resource Group.
  - tags - (Optional) A mapping of tags which should be assigned to the Resource Group.

3) Now seeing map (string) ,map(list) and map(object).

4) map (string) and map(list) is shown below.

```
type = map(string)
```

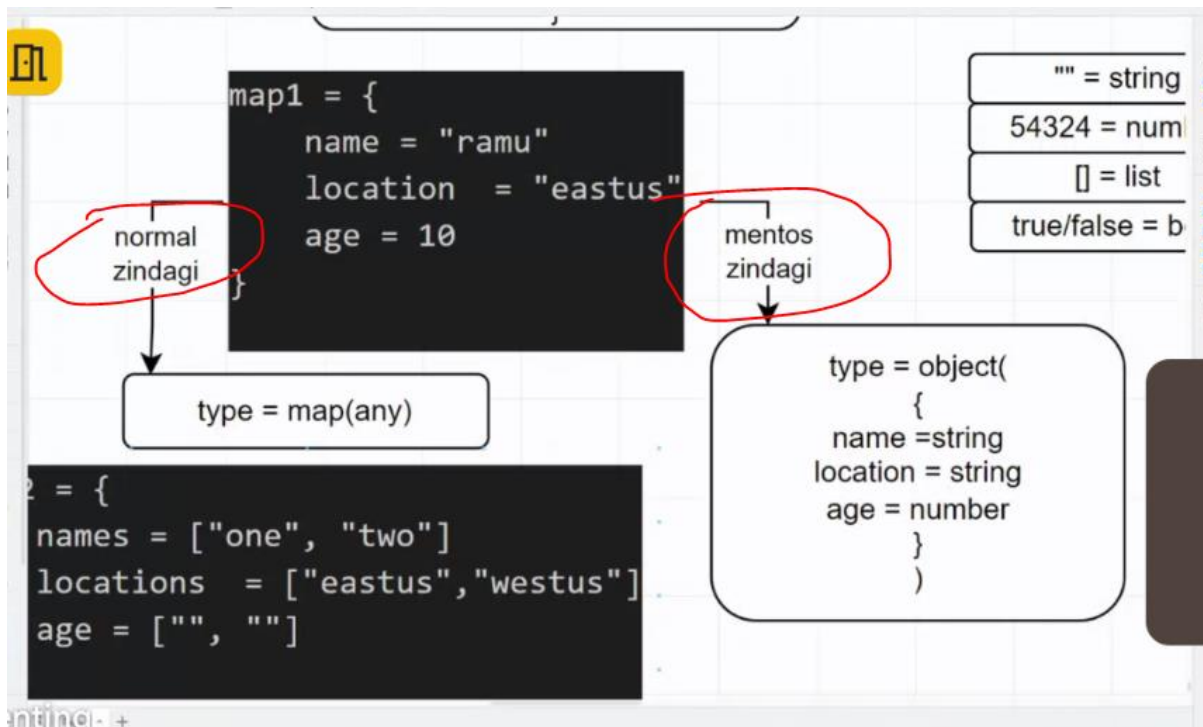
```
map2 = {  
  names = ["one", "two"]  
  locations = ["eastus", "westus"]  
  age = ["", ""]  
}
```

```
type = map(list(string))
```

```
map2 = {  
  rg1 = {
```

5) When we use `map(object)` then we specify every attribute differently i.e. `name = string`, etc

```
type = map(object(  
  {  
    name = string  
    location = string  
    age = number  
  })
```



6) Now writing same code by defining variable content in different types. We use optional attribute also

```

1 variable "rg_map" {
2   type = map(object(
3     {
4       name = string
5       location = string
6       tags = optional(map(string))
7       managed_by = optional(string)
8     }
9   ))
10 }
11
12 resource "azurerm_resource_group" "rg" {
13   for_each = var.rg_map
14   name     = each.value.name
15   location = each.value.location
16   tags     = each.value.tags
17 }

```

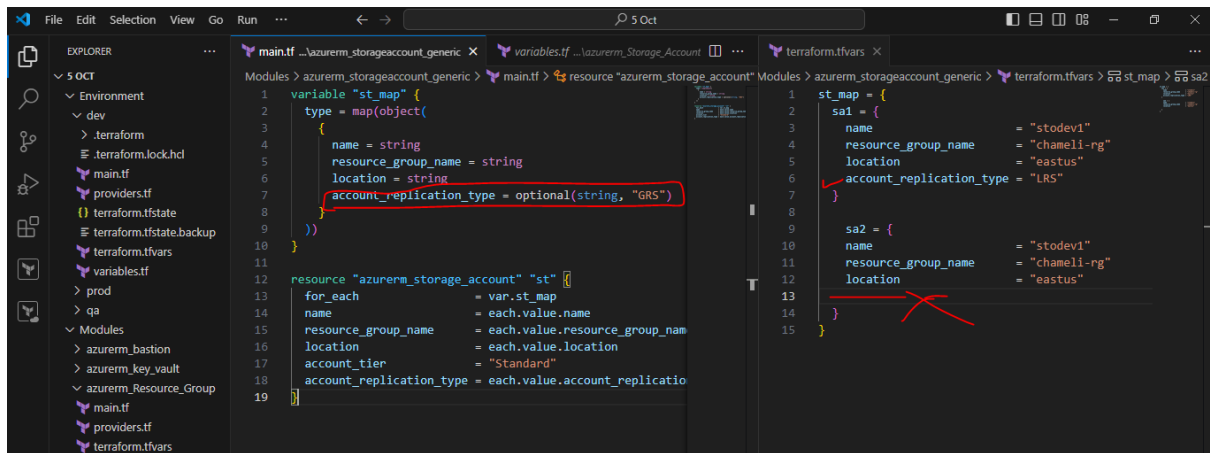
```

1 rg_map = {
2   rg1 = {
3     name     = "chameli-rg"
4     location = "eastus"
5     managed_by = "terraform"
6     tags = {
7       env = "dev"
8       company = "dhondhu"
9     }
10   }
11   rg2 = {
12     name     = "chameli-rg"
13     location = "eastus"
14   }
15 }
16

```

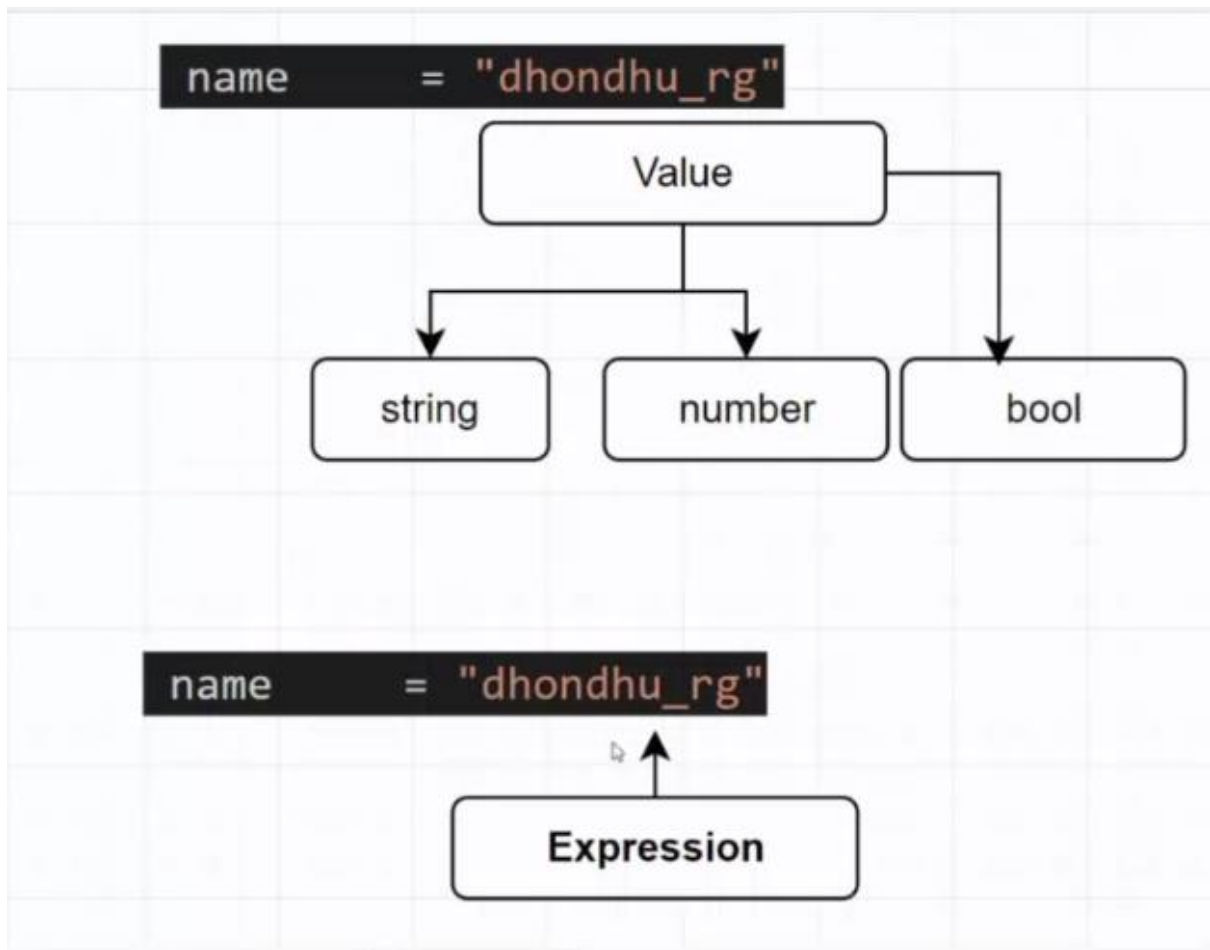
## +++++ AGENDA – Conditional dynamic block using in storage account using optional network rules

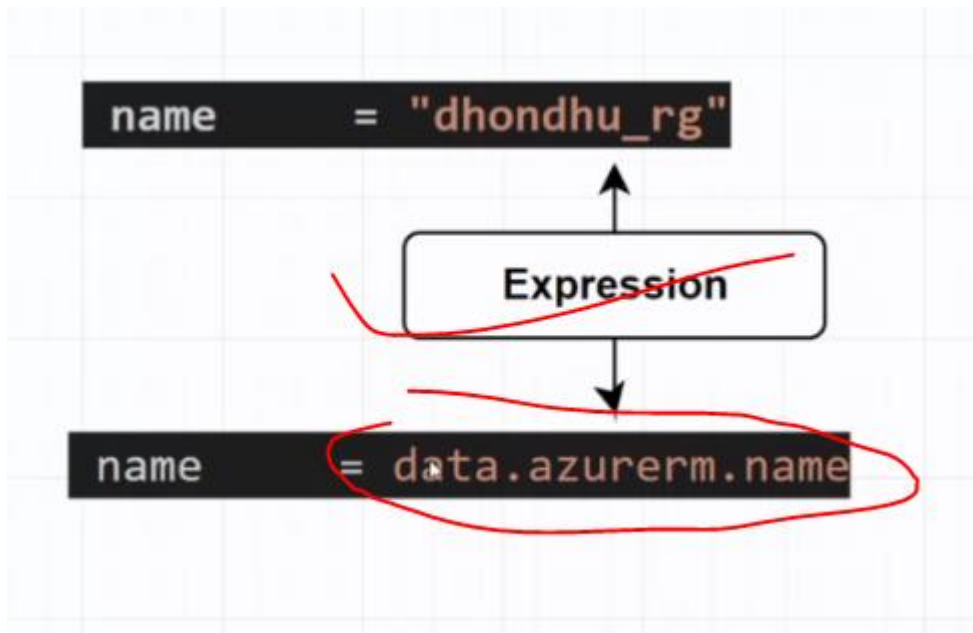
1) Now if someone do not passes account replication type value then we can straight way mention in variable in account replication type attribute. Then it will automatically take the value of GRS. The code will not fail if some user do pass empty value.



2) Network rules =

3) Whatever come after equal to is not value, its actually an expression.





4) Interpolation in terraform.

## Interpolation

A `${ ... }` sequence is an *interpolation*, which evaluates the expression given between the markers, converts the result to a string if necessary, and then inserts it into the final

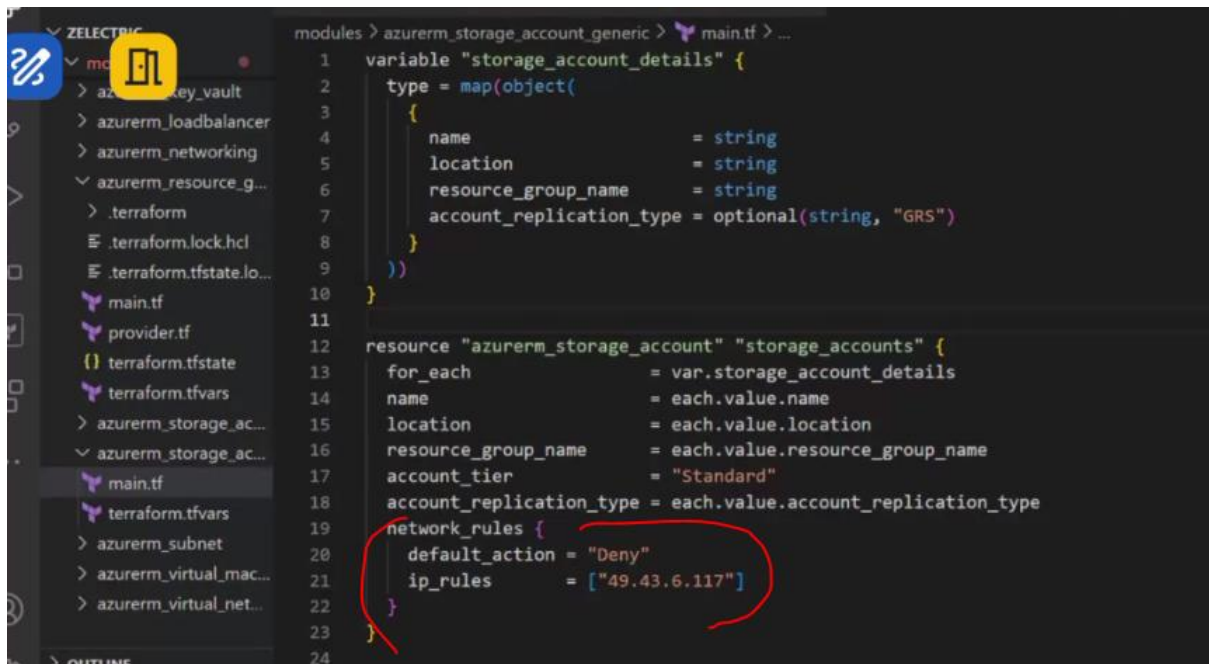
```
"Hello, ${var.name}!"
```

In the above example, the named object `var.name` is accessed and its value inserted.

```
1 resource "azurerm_resource_group" "rg" {  
2   name = "${var.vm_name}-nic"  
3   location = "westus"  
4 }  
5
```

5)

**ASSIGNMENT**



```
1 variable "storage_account_details" {
2   type = map(object(
3     {
4       name           = string
5       location       = string
6       resource_group_name = string
7       account_replication_type = optional(string, "GRS")
8     }
9   ))
10 }
11
12 resource "azurerm_storage_account" "storage_accounts" {
13   for_each = var.storage_account_details
14   name     = each.value.name
15   location = each.value.location
16   resource_group_name = each.value.resource_group_name
17   account_tier        = "Standard"
18   account_replication_type = each.value.account_replication_type
19   network_rules {
20     default_action = "Deny"
21     ip_rules       = ["49.43.6.117"]
22   }
23 }
24
```

Network rules wala sa me optional wala pata krke aao ki pass kre to chale na pass kre to na chale