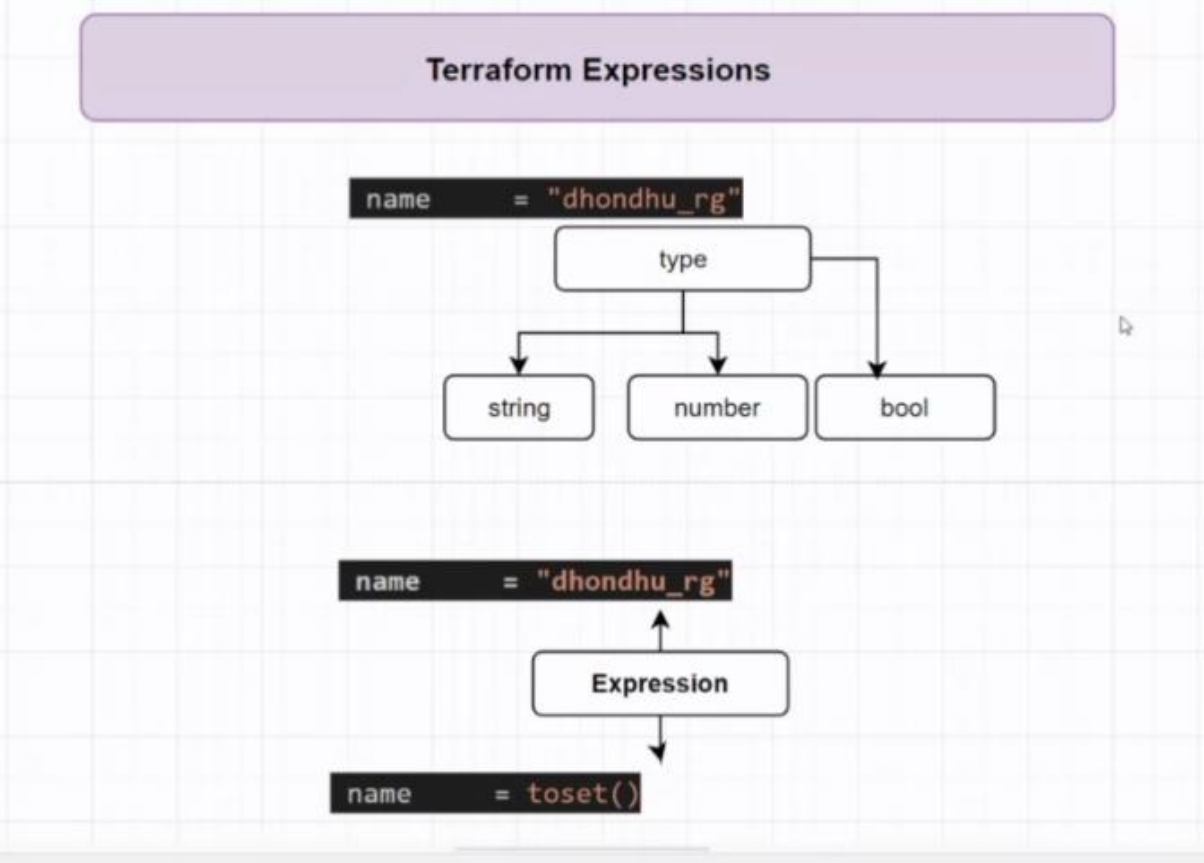
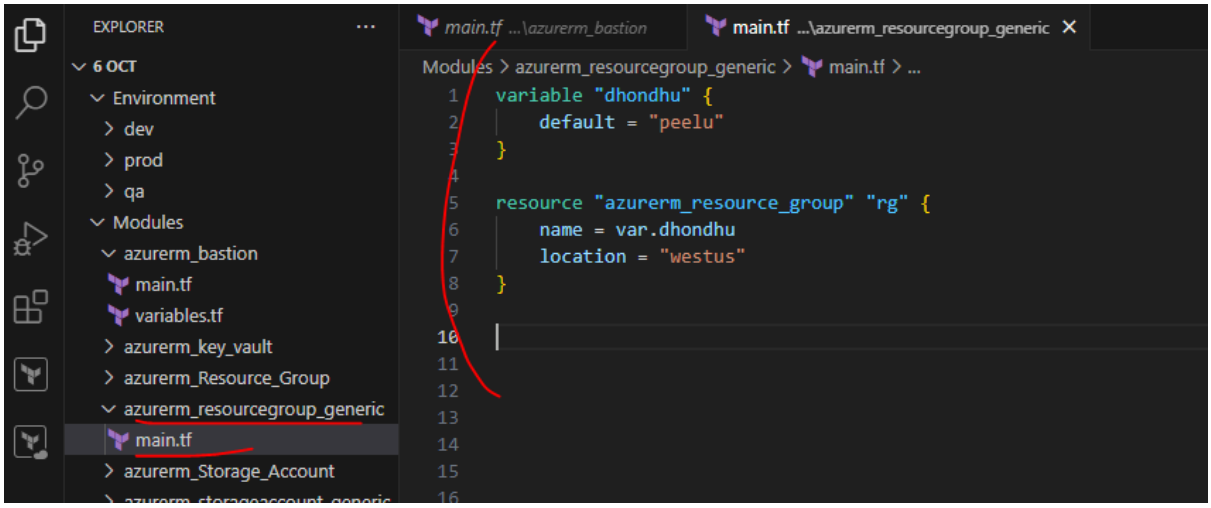


AGENDA – Ternary operator

1) SEARCH – Terraform expressions

2) Create “azurerm_resourcegroup_generic” folder and create main.tf file into it.



3) Now suppose, we have requirement to bring name in output after terraform plan as

i) rg-peelu

ii) rg-ramu

4) So for doing above we have to do string interpolation

```
main.tf ...\azurerm_bastion        main.tf ...\azurerm_resourcegroup_generic ×
Modules > azurerm_resourcegroup_generic > main.tf > ...
1  variable "dhondhu" {
2      default = "tony"
3  }
4
5  resource "azurerm_resource_group" "rg" {
6      name = "rg-${var.dhondhu}"
7      location = "westus"
8  }
9
```

5) Now for numbers

```
> resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 1 + 1
  location = "westus"
}
```

```
PORTS   AZURE   DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL

      + id       = (known after apply)
      + location = "westus"
      + name     = "2"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can'
```

```
main.tf
resource "azurerm_resource_group" "rg" > name
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 1 * 10
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
    + id       = (known after apply)
    + location = "westus"
    + name     = "10"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
resource "azurerm_resource_group" "rg" {
  name     = 79 % 10
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
    + id       = (known after apply)
    + location = "westus"
    + name     = "9"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

6) Equality operators

Equality Operators

The equality operators both take two values of any type and produce boolean values as results.

- `a == b` returns `true` if `a` and `b` both have the same type and the same value, or `false` otherwise.
- `a != b` is the opposite of `a == b`.

Because the equality operators require both arguments to be of exactly the same t
to decide equality, we recommend using these operators only with values of primit
using explicit type conversion functions to indicate which type you are intending to
comparison.

```
main.tf
resource "azurerm_resource_group" "rg" > name
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 79 == 79
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
    + id       = (known after apply)
    + location = "westus"
    + name     = "true"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
main.tf
resource "azurerm_resource_group" "rg" > name
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 79 == 80
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
  + id       = (known after apply)
  + location = "westus"
  + name     = "false"
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

7) Now terraform has ability to compare strings also as well

```
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = "dhondhu" == "dhondhu"
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
  + id       = (known after apply)
  + location = "westus"
  + name     = "true"
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
main.tf
resource "azurerm_resource_group" "rg" > location
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = "dhondhu" == "tondu"
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
    + id       = (known after apply)
    + location = "westus"
    + name     = "false"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

8) "rg-${79==79}-${var.dhondhu}"

```
main.tf
resource "azurerm_resource_group" "rg" > name
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = "rg-${79==79}-${var.dhondhu}"
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
    + id       = (known after apply)
    + location = "westus"
    + name     = "rg-true-tony"
  }
```

9) 79 != 79

```
main.tf   ×   provider.tf
> resource "azurerm_resource_group" "rg"
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 79 != 79
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
    + location = "westus"
    + name     = "false"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
main.tf   ×   provider.tf
> resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 79 != 80
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
  + resource "azurerm_resource_group" "rg" {
      + id       = (known after apply)
      + location = "westus"
      + name     = "true"
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

10)

Logical Operators

The logical operators all expect bool values and produce bool values as results.

- `a || b` returns `true` if either `a` or `b` is `true`, or `false` if both are `false`.
- `a && b` returns `true` if both `a` and `b` are `true`, or `false` if either one is `fal`
- `!a` returns `true` if `a` is `false`, and `false` if `a` is `true`.

Terraform does not have an operator for the "exclusive OR" operation. If you know

i) true||false, false||false

main.tf × provider.tf

resource "azurerm_resource_group" "rg" > name

```
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = true || false
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
  + location = "westus"
  + name     = "true"
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
resource "azurerm_resource_group" "rg" {
    name = false||false
    location = "westus"          #Answer = false
}
```

ii) true&&false, true&&true

```
resource "azurerm_resource_group" "rg" {
    name = true&&false
    location = "westus"     #Answer = false
}

resource "azurerm_resource_group" "rg" {
    name = true&&true
    location = "westus"    #Answer = true
}
```

iii) 79==79 && 78!=79

```
resource "azurerm_resource_group" "rg" {
  name     = 79==79 && 78!=79
  location = "westus"
}
true && true
true
```

12) Conditional Expressions

Syntax

The syntax of a conditional expression is as follows:

```
condition ? true_val : false_val
```

If `condition` is `true` then the result is `true_val`. If `condition` is `false` then the result is `false_val`.

A common use of conditional expressions is to define defaults to replace invalid values:

```
var.a != "" ? var.a : "default-a"
```

If `var.a` is an empty string then the result is `"default-a"`, but otherwise it is the actual value of `var.a`.

```
resource "azurerm_resource_group" "rg" {
    name = condition ? true_val : false_val
    location = "westus"    #Answer = true
```

```
}
```

i) true ? "dhondhu" : "tondu"

```
> resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = true ? "dhondhu" : "tondu"
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
+ resource "azurerm_resource_group" "rg" {
    + id       = (known after apply)
    + location = "westus"
    + name     = "dhondhu"
  }
```

ii) false ? "dhondhu" : "tondu"

```
main.tf   ×   provider.tf
> resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = false ? "dhondhu" : "tondu"
  location = "westus"
}
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
+ resource "azurerm_resource_group" "rg" {
    + id       = (known after apply)
    + location = "westus"
    + name     = "tondu"
  }
```

i) 79==79 ? "dhondhu" : "tondu"

```
main.tf   ×   provider.tf
> resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 79==79 ? "dhondhu" : "tondu"
  location = "westus"
}

PORTS   AZURE   DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL
  + location = "westus"
  + name     = "dhondhu"
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

ii) 78==79 ? "dhondhu" : "tondu"

```
main.tf   provider.tf
> resource "azurerm_resource_group" "rg" > name
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = 78==79 ? "dhondhu" : "tondu"
  location = "westus"
}

PORTS   AZURE   DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL

+ resource "azurerm_resource_group" "rg" {
    + id       = (known after apply)
    + location = "westus"
    + name     = "tondu"
  }
```

i) var.dhondhu == "tony" && 64329!=64328 ? "dhondhu" : "tondu"

```
main.tf   provider.tf
> resource "azurerm_resource_group" "rg"
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = var.dhondhu == "tony" && 64329!=64328 ? "dhondhu" : "tondu"
  location = "westus"
}

PORTS   AZURE   DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
    + id       = (known after apply)
    + location = "westus"
    + name     = "dhondhu"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

ii) var.dhondhu == "tony1" && 64329!=64328 ? "dhondhu" : "tondu"

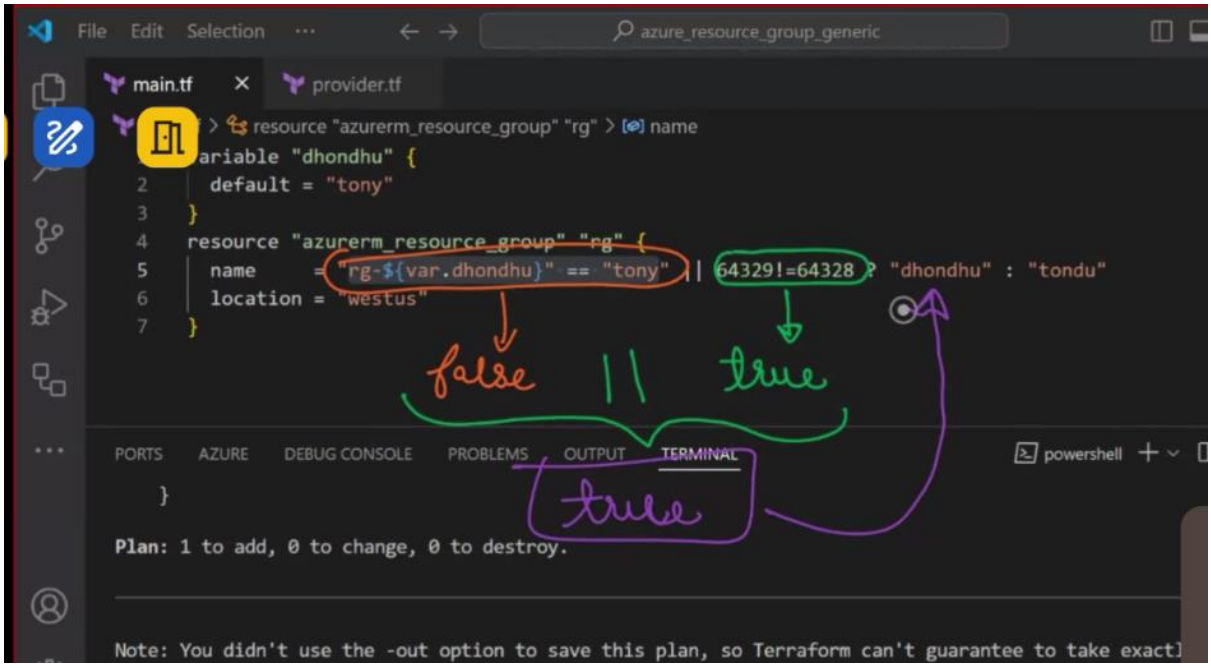
```
main.tf   provider.tf
resource "azurerm_resource_group" "rg" > name
ariable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = var.dhondhu == "tony1" && 64329!=64328 ? "dhondhu" : "tondu"
  location = "westus"
}

PORTS   AZURE   DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL

    + location = "westus"
    + name     = "tondu"
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

i) $ means value

"rg-${var.dhondhu}" == "tony" && 64329!=64328 ? "dhondhu" : "tondu"

```
resource "azurerm_resource_group" "rg" {
    name = "rg-${var.dhondhu}" == "tony" && 64329!=64328 ? "dhondhu" : "tondu"
    location = "westus"                    #Answer = tondu
}
```

ii) "rg-${var.dhondhu}" == "tony" || 64329!=64328 ? "dhondhu" : "tondu"



```
variable "dhondhu" {
  default = "tony"
}
resource "azurerm_resource_group" "rg" {
  name     = "rg-${var.dhondhu}" == "tony" || 64329!=64328 ? "dhondhu" : "tondu"
  location = "westus"
}
```

```
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactl
```

++

AGENDA – Making VM

++

AGENDA – Count loop

1)

2) Now suppose we have to delete 3rd rg but it will delete 4th one as well. We can see below in terraform plan

i) count =3

```
main.tf   provider.tf
resource "azurerm_resource_group" "rg" > # count
resource "azurerm_resource_group" "rg" {
  count    = 3
  name     = "rg-tony-${count.index+1}"
  location = "westus"
}

# resource "azurerm_resource_group" "rg1" {
#   for_each = ["rg-tony-1","rg-tony-2","rg-tony-4","rg-tony-5"]
#   name     = each.value
#   location = "westus"
```

PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
Plan: 0 to add, 0 to change, 2 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guara
if you run "terraform apply" now.
```

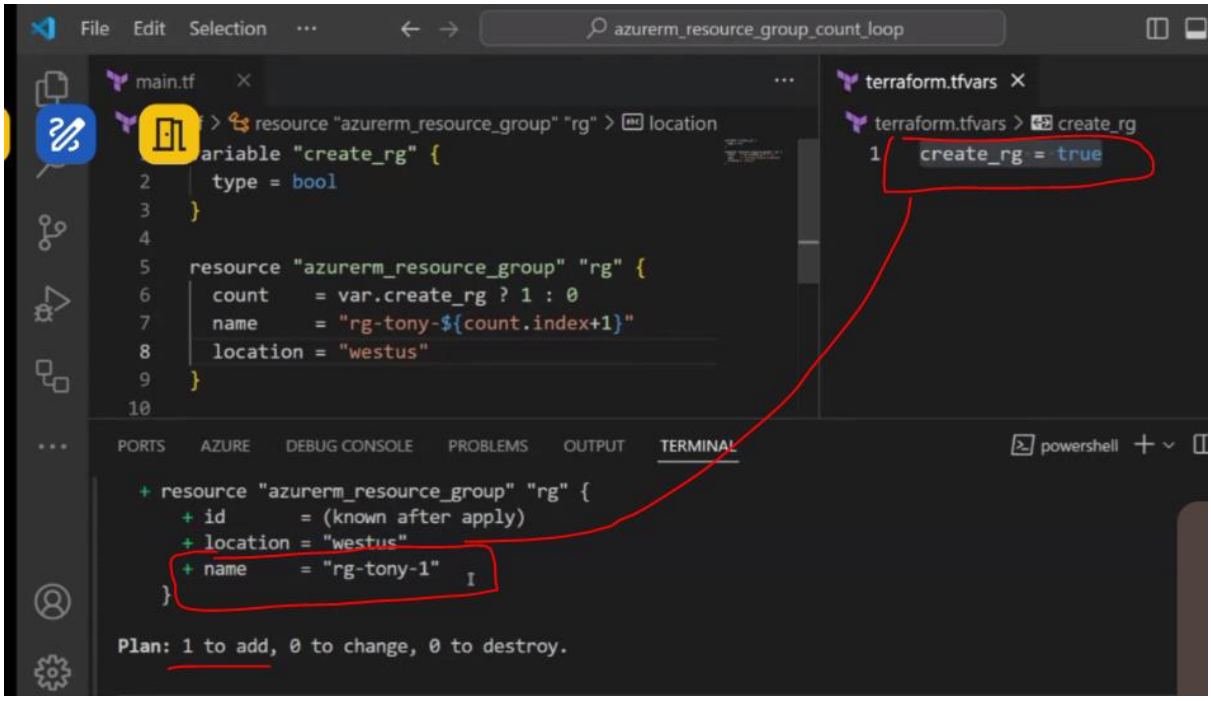
ii) count = 0

```
main.tf   provider.tf
resource "azurerm_resource_group" "rg"
resource "azurerm_resource_group" "rg" {
  count    = 0
  name     = "rg-tony-${count.index+1}"
  location = "westus"
}
```

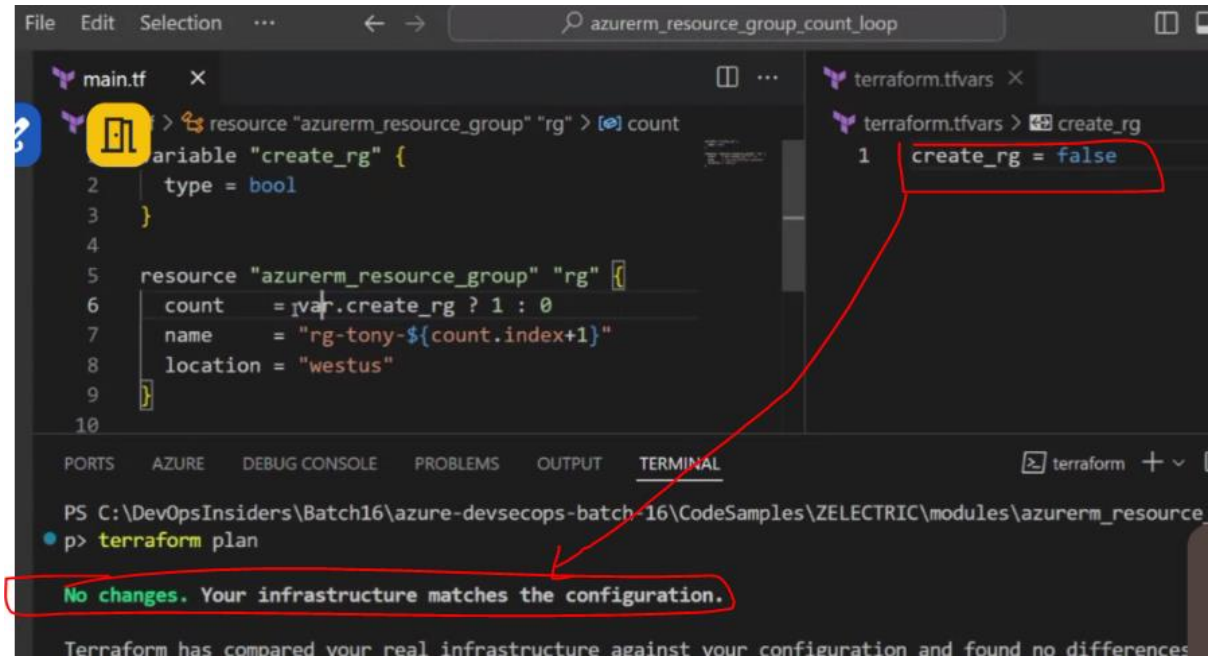
PORTS AZURE DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
PS C:\DevOpsInsiders\Batch16\azure-devsecops-batch-16\CodeSamples\ZEL
p> terraform apply

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configura
are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\DevOpsInsiders\Batch16\azure-devsecops-batch-16\CodeSamples\ZEL
```

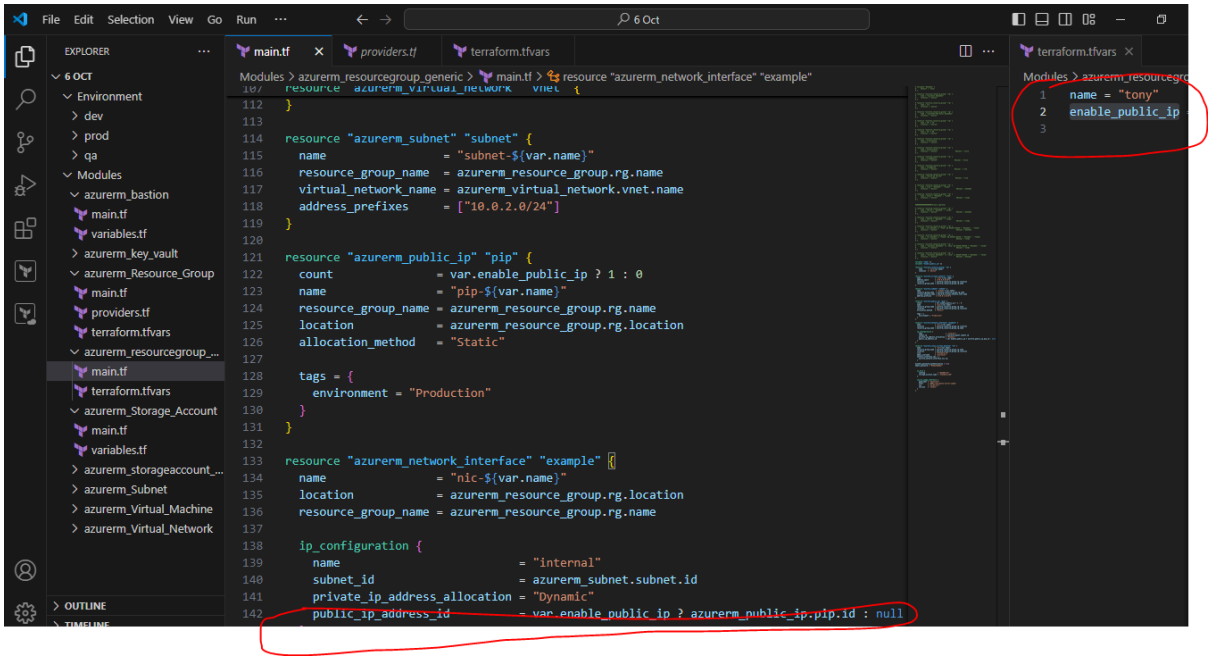
i) In terraform.tfvars when passed create-rg value as true.



ii) In terraform.tfvars when passed create-rg value as false.



i)



```
resource "azurerm_virtual_network" "vnet" {
}

resource "azurerm_subnet" "subnet" {
  name                 = "subnet-${var.name}"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.0.2.0/24"]
}

resource "azurerm_public_ip" "pip" {
  count               = var.enable_public_ip ? 1 : 0
  name                = "pip-${var.name}"
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  allocation_method   = "Static"

  tags = {
    environment = "Production"
  }
}

resource "azurerm_network_interface" "example" {
  name                = "nic-${var.name}"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                          = "internal"
    subnet_id                     = azurerm_subnet.subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          = var.enable_public_ip ? azurerm_public_ip.pip.id : null
```

```
name = "tony"
enable_public_ip
```

++