Day 63 - Terraform Variables

variables in Terraform are quite important, as you need to hold values of names of instance, configs, etc.

There are Three ways, you can set a value to a variable:

1] **Directly in main.tf files** (Varabiles block are included inside main.tf file)

2] **Set variable value on variable.tf file.** (Main.tf,Variable.tf).

Main.tf: - mention provider block and resource block

Varaible.tf :- Setting the default value of a variable.

3] **Using Command Line**

Create 2 files (Main.tf,Variable.tf), but inside variable.tf, don’t set the default value.

Pass the value of a variable through command line.

We can create a variables.tf file which will hold all the variables.

```
variable "file"{
default = "/home/ubuntu/TerraForm/demo.txt"
}

variable "content"{
default = "This is coming from Variable.tf files"
}
```

These variables can be accessed by var object in main.tf

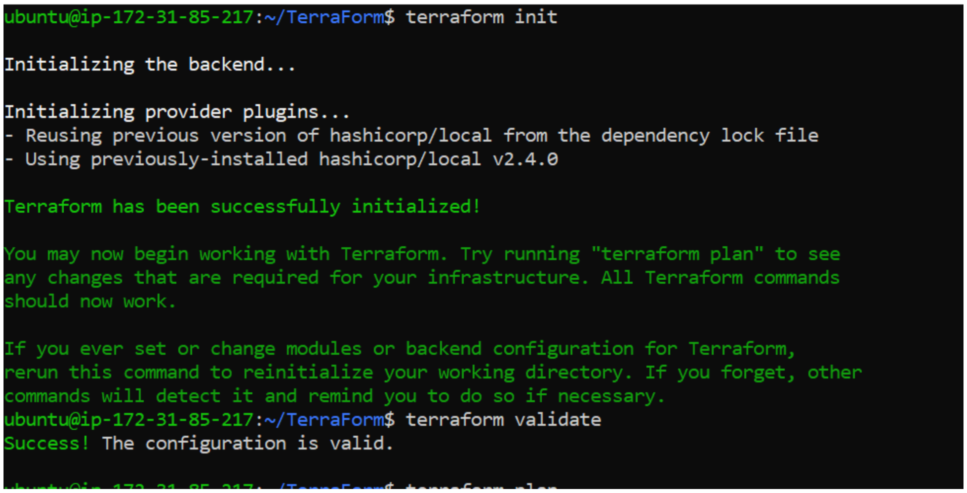
Task-01

- Create a local file using Terraform Hint:

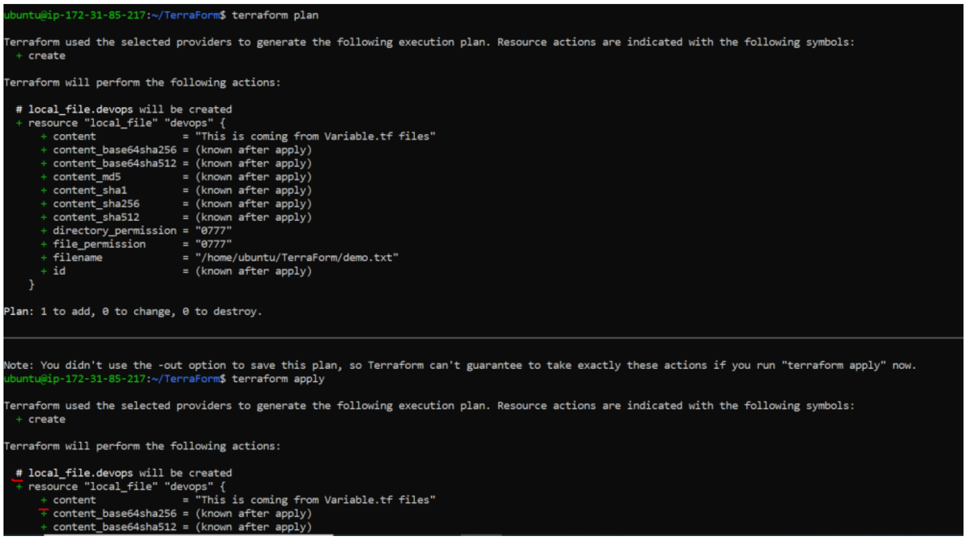
```
resource "local_file" "devops"{
filename = var.file
content = var.content
}
```

After all changes are done, run the files using below four commands

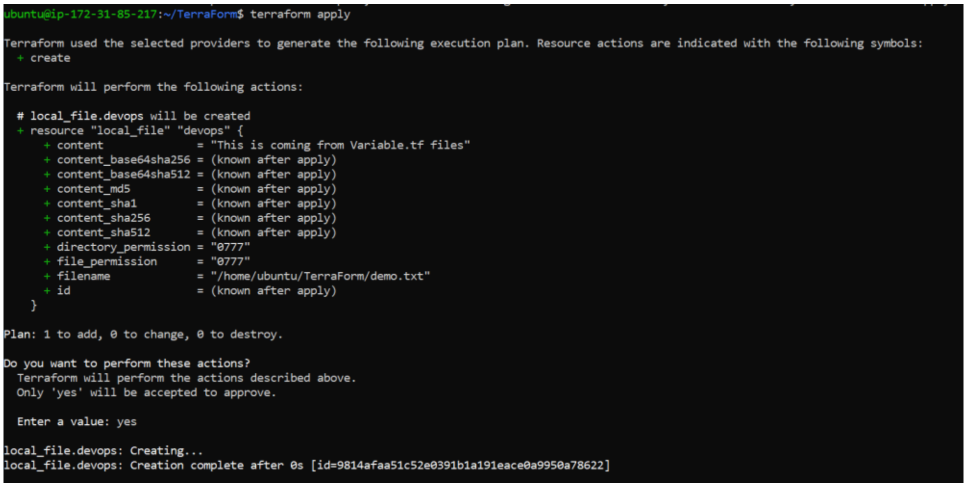
```
terraform init
terraform validate
terraform plan
terraform apply
```



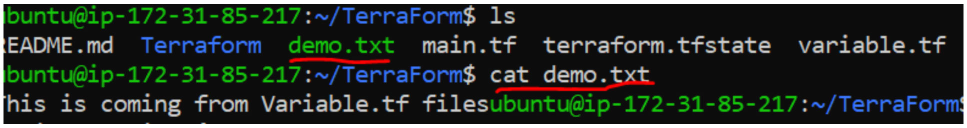
```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.4.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-85-217:~/TerraForm$ terraform plan
```



```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.devops will be created
  + resource "local_file" "devops" {
      + content              = "This is coming from Variable.tf files"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "/home/ubuntu/TerraForm/demo.txt"
      + id                   = (known after apply)
    }

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 exactly these actions if you run "terraform apply" now.
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.devops will be created
  + resource "local_file" "devops" {
      + content              = "This is coming from Variable.tf files"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
```



```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.devops will be created
  + resource "local_file" "devops" {
      + content              = "This is coming from Variable.tf files"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "/home/ubuntu/TerraForm/demo.txt"
      + id                   = (known after apply)
    }

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

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.devops: Creating...
local_file.devops: Creation complete after 0s [id=9814afaa51c52e0391b1a191eace0a9950a78622]
```



```
buntu@ip-172-31-85-217:~/TerraForm$ ls
EADME.md  Terraform  demo.txt  main.tf  terraform.tfstate  variable.tf
buntu@ip-172-31-85-217:~/TerraForm$ cat demo.txt
his is coming from Variable.tf filesubuntu@ip-172-31-85-217:~/TerraForm
```

Task-02

- Use terraform to demonstrate usage of List, Set and Object datatypes

Data Types in Terraform

1] List: - In a same variable, you can add multiple values separated by comma (,).

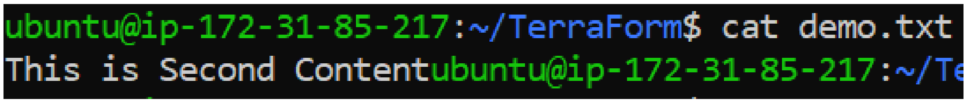
Modifying the Variable.tf file

Set type= list and list out all values, index starts with 0.

```
variable "file"{
default = "/home/ubuntu/TerraForm/demo.txt"
}

variable "contents"{
type = list
default = ["This is First Content","This is Second Content"]
}
```

Main.tf

```
resource "local_file" "devops"{
filename = var.file
content = var.contents[1]
}
```



```
ubuntu@ip-172-31-85-217:~/TerraForm$ cat demo.txt
This is Second Contentubuntu@ip-172-31-85-217:~/Te
```

2] Map: - In a same variable, you can add data as a key-value pair.

Modifying the Variable.tf file

Set type= map and add data as a key-value pair.

```
variable "file"{
default = "/home/ubuntu/TerraForm/demo.txt"
}

variable "contents"{
type = map
default = {
"Content1" = "This is First Content",
"content2" = "This is Second Content"
}
}
```

Main.tf

```
resource "local_file" "devops"{
filename = var.file
content = var.contents["Content1"]
}
```

3] Set

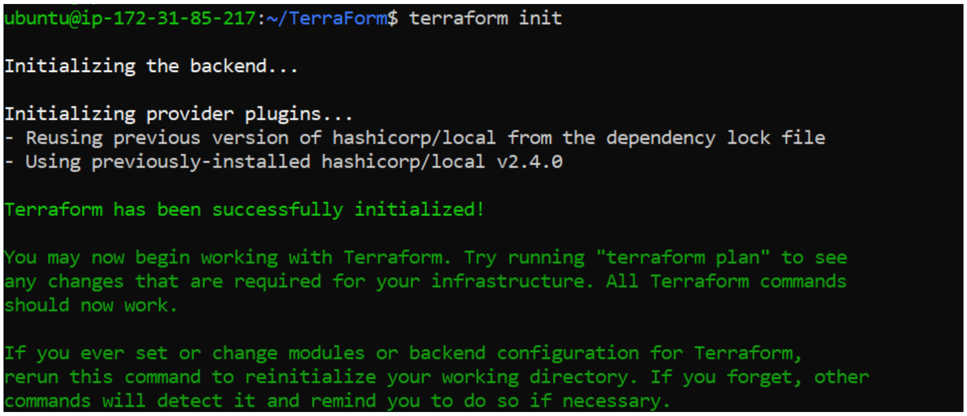
4] Object: - An object is a structural type that can contain different types of values, unlike map, list. It is a collection of named attributes that each have their own type.

Variable.tf

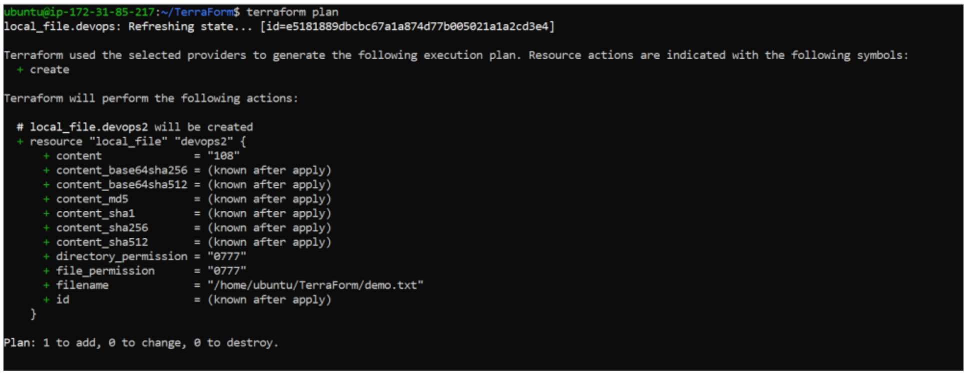
```
variable "file"{
default = "/home/ubuntu/TerraForm/demo.txt"
}

variable "contents"{
type = object ({
Content1 = string
content2 = number
})
default = {
"Content1" = "This is First Content",
"content2" = 108
}
}
```

Main.tf

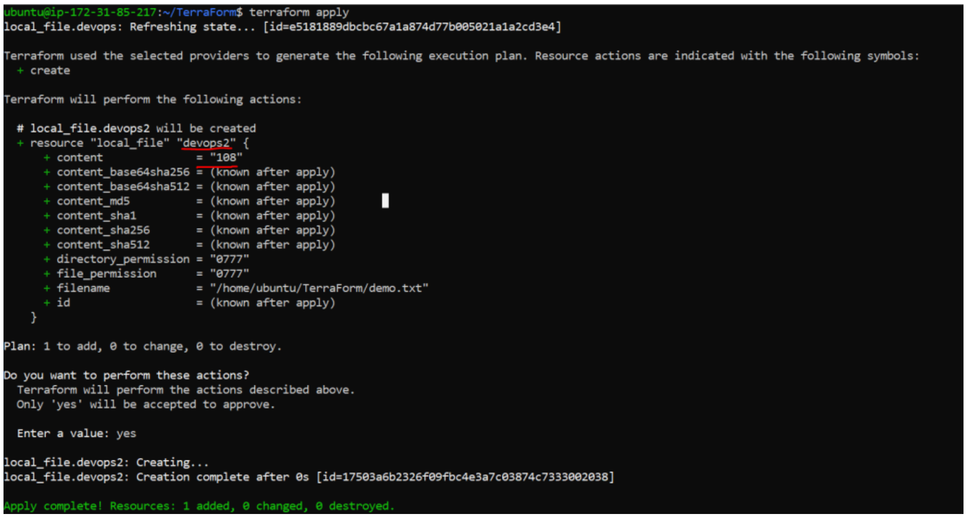
```
resource "local_file" "devops"{
filename = var.file
content = var.contents["Content1"]
}


resource "local_file" "devops2"{
filename = var.file
content = var.contents["content2"]
}
```



```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.4.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```



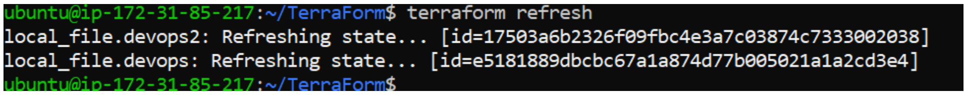
```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform plan
local_file.devops: Refreshing state... [id=e5181889dbcbc67a1a874d77b005021a1a2cd3e4]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.devops2 will be created
  + resource "local_file" "devops2" {
      + content              = "108"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "/home/ubuntu/TerraForm/demo.txt"
      + id                   = (known after apply)
    }

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



```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform apply
local_file.devops: Refreshing state... [id=e5181889dbcbc67a1a874d77b005021a1a2cd3e4]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.devops2 will be created
  + resource "local_file" "devops2" {
      + content              = "108"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "/home/ubuntu/TerraForm/demo.txt"
      + id                   = (known after apply)
    }

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

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.devops2: Creating...
local_file.devops2: Creation complete after 0s [id=17503a6b2326f09fbc4e3a7c03874c7333002038]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Use terraform refresh

To refresh the state by your configuration file, reloads the variables



```
ubuntu@ip-172-31-85-217:~/TerraForm$ terraform refresh
local_file.devops2: Refreshing state... [id=17503a6b2326f09fbc4e3a7c03874c7333002038]
local_file.devops: Refreshing state... [id=e5181889dbcbc67a1a874d77b005021a1a2cd3e4]
ubuntu@ip-172-31-85-217:~/TerraForm$
```